A PATH ENUMERATION ALGORITHM FOR DETERMINING THE COMPLETION TIME OF PERT NETWORKS

A Thesis Submitted
In Partial Fulfilment of the Requirements
for the Degree of
MASTER OF TECHNOLOGY

By
S. S. R. MURTHY

to the

INDUSTRIAL AND MANAGEMENT ENGINEERING PROGRAMME
INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

FEBRUARY, 1985

CERTIFICATE

This is to certify that the present work on "A Path Enumeration Algorithm for Determining the Completion Time of PERT Networks", by S.S.R. Murthy has been carried out under my supervision and has not been submitted elsewhere for the award of a degree.

(R.K. Ahuja)
Lecturer
Industrial and Management Engg.
Indian Institute of Technology
Kanpur 208016, INDIA

February, 1985

IMEP-1985-M-MUR-PAT

ACKNOWLEDGEMENTS

I express my deep sense of gratitude to my guide, Dr. R. K. Ahuja, who has provided me his invaluable suggestions, undescribable help and unforgettable attachment during the course of this work.

I am very thankful to my friend, Mr. V.V.S.N. Murty, for his patient hearing of my problems while carrying out the work and for his timely help. I am also thankful to Mr. S.V.R. Reddy and other friends for their ready willingness to help me in any sense at any time.

Lastly, I thank Mr. J.K. Misra for his neat and accurate typing, Mr. D.K. Misra for his appreciable plotting of graphs, and Mr. Buddhi Ram Kandiyal for his immaculate cyclostyling work.

S.S.R. Murthy

CONTENTS

ABSTRACT

In this thesis, the problem of finding the mean and standard deviation of the project duration in PERT networks is considered. The conventional method, which assumes the expected values of activity durations as actual times for completion, always gives lowest bound to the project duration. Monte Carlo Simulation technique, which can give fairly accurate results, takes very high time even for networks of reasonable size. A new approximate method is developed in this work which appears to be a reasonable compromise between the two extremes. The method enumerates k-th longest paths in the network and finds the maximum of corresponding random variables. Dependency among the paths is also considered in an effective way. The suggested method is extensively tested on randomly generated PERT networks with various topological structures. The results are compared with conventional method, simulation method and Sculli's method. The results of the computational analysis, which are found to be quite encouraging, are also presented.

CHAPTER I

INTRODUCTION

1.1 INTRODUCTION:

One of the most important problems in the analysis of PERT networks is the determination of the total completion time of the project. When the completion times of activities of the project are random variables, the project duration is also a random variable with a distribution function that is a complex function of the corresponding distribution functions of each activity. Because of this complexity in evaluating the distribution function, the calculation of mean and standard deviation of the project duration has become a difficult problem and drawn the attention of many researchers. There is no exact method or formulae available in the literature for finding the mean and standard deviation of project duration and one has to content with approximate methods.

The conventional method of solving the problem assumes the time of each activity as equal to its expected value and treats the network as a deterministic one. It is easy to realise that this gives only the crude lower bound on the project duration. Monte Carlo Simulation can be used to get fairly accurate results but it takes a very high time even for networks of reasonable size. Hence the research work has been directed to develop

approximate methods which improve the results of conventional method in a reasonable amount of time. Some of the existing approximate methods are due to Fulkerson [2], Elmaghraby [4], Robillard and Trahan [6], Sculli [8] etc. Though these methods are able to improve the accuracy to some extent, there is still a great scope for work to further improve the accuracy in the mean and standard deviation of project duration.

This thesis work is directed to developing a new approximate method for the problem which gives tolerably accurate results even at an acceptable increase in the computational time.

1.2 PRELIMINARIES:

The following terminology, notations and statistical rules are used quite often in the thesis work. For the sake of completeness, they are given below.

1.2.1 <u>Network Notations</u>:

A directed network G(N, A) consists of a finite set N of elements, called <u>nodes</u>, and a set A of ordered pairs of nodes, called <u>arcs</u>, with some weights attached to the nodes and arcs.

Let, $n = |N|$ = no. of nodes in G

$m = |A|$ = no. of arcs in G

If there is an arc e from node i to node j, then i is called <u>Tail node</u> of the arc e and j is called <u>Head node</u>. Also, the arc e is said to be <u>incident from</u> node i and <u>incident to</u>

node j. Let I_i and O_i denote, respectively, the sets of arcs incident to and incident from node i. The node $S \in N$ which has no arcs incident to it, is called the Source and the node $t \in N$ which has no arcs incident from it, is called the Sink.

A Path in $G = (N, A)$ is a sequence $i_1, i_2, \ldots, i_r$ of distinct nodes of N such that either $(i_k, i_{k+1}) \in A$ or $(i_{k+1}, i_k) \in A$ for each $k = 1, 2, \ldots, r-1$. A directed Path is defined similarly, except that $(i_k, i_{k+1}) \in A$ for each $k = 1, 2, \ldots, r-1$. A cycle is a path together with an arc (i_r, i_1) or (i_1, i_r). A directed cycle is a directed path together with the arc (i_r, i_1).

A network is said to be acyclic if it does not contain any cycle. Topological Sorting is a systematic way of finding the existence of a cycle in the given network. The procedure for topological sorting is as follows:

Step 0: Set $i = 0$

Step 1: Set $i = i+1$

Identify a node with no incoming arcs and number it as i.

Delete the node i and its outgoing arcs.

If i = 't' then STOP, else go to Step 1.

It is obvious from the above procedure that all nodes can be numbered if and only if there is no directed cycle in the network.

A directed complete network is one in which for every node $i \in N$, $(i, k) \in A$ for all $k = i+1, \ldots, n$. The Length of a path in G is the sum of the lengths of all the arcs on that path.

1.2.2 PERT Terminology:

Any project consists of a number of job operations or tasks to be performed which are called activities. An Event is a specific instant of time which marks the beginning and the ending of an activity.

Network representation of a project is very useful in Project Management because it gives a clear picture of the project in a snapshot. In this representation, the activities of the project correspond to the arcs, the events to the nodes and the activity durations are the weights attached to the arcs. The position of an activity is dictated by the following rule: all the activities associated with the incoming arcs to a node must be terminated before those associated with the outgoing arcs at the node can start. In consequence, a project network can be only directed acyclic.

If the durations of all the activities can be predicted exactly before starting the project, then the network is said to be a deterministic network. On the other hand, if the activity durations cannot be predicted exactly due to some chance variations, then the network becomes a probabilistic network or

PERT (Program Evaluation and Review Technique). This is often the case of research and development activities, projects of non-repetitive nature such as defence projects.

Project duration is the total time required to complete the project. The sequence of activities which decide the project duration form the critical path. The path for which the length is nearly equal to the critical path length is called a near critical path.

1.2.3 Probability Theory in PERT:

The probability theory and mathematical statistics have found a great use in the analysis of PERT networks. One of their most important uses in PERT network analysis is to get a measure of the uncertainty that the project will be completed before a specified due date. The statements of the possible range of times for the completion of the project and the probabilities associated with each are quite useful for the management. By adding to this information an appraisal of the consequences of not meeting a scheduled date, the management can better plan a project.

Normal distribution is often referred in the thesis work. It is a symmetrical bell shaped curve with two parameters, mean (m) and standard deviation (σ). The probability density function for a normally distributed random variable X is given by,

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-m}{\sigma}\right)^2}, \quad -\infty < x < \infty$$

The cumulative distribution function, $F(x)$, involves the integration of the above function, between the limits $-\infty$ and x, which is difficult to evaluate.

The normal distribution with the parameters $m = 0$ and $\sigma = 1$ is called Standard Normal distribution. Its density function and distribution function, denoted by $\psi(x)$ and $\phi(x)$ respectively, are given by,

$$\psi(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2},$$

and $$\phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{1}{2}x^2} dx.$$

The importance of $\phi(x)$ is that it gives the cumulative area upto the point x in the Standard Normal Curve, i.e. it gives the probability for an observed value of X to be less than or equal to x. The values of $\phi(x)$ are tabulated for different values of x and are made available in any statistical book.

As a general case, if one wants to compute Prob $[x \leq x]$, where X is Normally distributed with parameters m and σ, then x can be converted to the Standard Normal Scale by $x = (x-m)/\sigma$ and the tables for that value of x can be seen. It is interesting to note that 99.7 percent of theobserved values for X lie within the range $(m - 3\sigma)$ and $(m + 3\sigma)$.

With the background about the Normal distribution, we now present some formulae and theorems available in the literature.

(i) Clark's Formulae [13]:

These are used for finding the mean and standard deviation of the maximum of two independent Normal random variables. Suppose X_1 and X_2 are two such variables with parameters (m_1, σ_1) and (m_2, σ_2) respectively. If $Y = \max(X_1, X_2)$, the parameters of Y, m and σ, are given by

$$m = m_1\,\phi(\alpha) + m_2\,\phi(-\alpha) + a\,\psi(\alpha)$$

$$V_2 = (m_1^2 + \sigma_1^2)\,\phi(\alpha) + (m_2^2 + \sigma_2^2)\,\phi(-\alpha) + (m_1 + m_2)\,a\,\psi(\alpha)$$

and, $\sigma = V_2 - m^2$

where, $a = \sigma_1^2 + \sigma_2^2$ and $\alpha = (m_1 - m_2)/a.$

(ii) Approximate Formula for $\phi(\alpha)$:

The following approximate formula given in Greer and Cacava [9] can be used to find $\phi(x)$ for a given value of x.

$$\phi(x) = 1 - \frac{1}{\sqrt{2\pi}}\, e^{-\frac{1}{2}x^2}\left[\frac{c_2}{(1+c_1x)^5} - \frac{c_3}{(1+c_1x)^4} + \frac{c_4}{(1+c_1x)^3} - \frac{c_5}{(1+c_1x)^2} + \frac{c_6}{(1+c_1x)}\right]$$

where, $c_1 = 0.2316419$, $c_2 = 1.330274$, $c_3 = 1.821256$
$c_4 = 1.781478$, $c_5 = 0.3565638$, $c_6 = 0.3193815$.

It is important to note that if x is negative, the above formula should not be used directly. Instead, it can be used for the absolute value of x and then use $\phi(-x) = 1-\phi(x)$. This formula gives $\phi(x)$ upto 5 digits of accuracy.

(iii) Jensen's Inequality [15]:

According to this inequality, the expected value of the maximum of a set of independent random variables is greater than or equal to the maximum of the expected values. Mathematically, if $X_1, X_2, \ldots, X_n$ are independent random variables,

$$E[\text{Max } (X_1, X_2, \ldots, X_n)] \geq \text{Max } [E(X_1), E(X_2), \ldots, E(X_n)]$$

Similarly, the variance of the maximum is, in general less than the minimum of the variances. This is particularly true when the individual distributions are symmetric such as Normal.

(iv) Central Limit Theorem:

This theorem is one of the most important theorems in all of mathematical statistics. According to this theorem, the distribution of the sum of n independent random variables is approximately Normal regardless of the distributions of individual random variables, if n is sufficiently large.

(v) Sums of Random Variables:

The mean of the sum of n independent Normal random variables is equal to the sum of the means of individual random

variables. Similarly, the variance of the sum is the sum of the variances.

Mathematically
$$E\,[\Sigma\, X_i] = \Sigma\, E\,(X_i)$$
$$\mathrm{Var}\,[\Sigma\, X_i] = \Sigma\, \mathrm{Var}\,(X_i)$$

1.3 MOTIVATION OF THE PROBLEM:

The problem of finding the project duration is very easy in deterministic networks as all the activity durations are fixed and well known. Infact, the project duration becomes the length of the longest path in the network.

The same problem, however, becomes exceedingly difficult in PERT networks. Since the activity durations are random variables, each path in the network from source to sink also becomes a random variable which is equal to the sum of all the random variables corresponding to the activities on that path. Hence project duration also becomes a random variable that is equal to the maximum of all the random variables corresponding to the paths. The mean and the standard deviation of project duration are difficult to evaluate because of two main reasons (i) the number of paths in the network may be intractable for large networks and (ii) we need to find the maximum of a set of dependent random variables, corresponding to the paths in the network, for which no method is available.

The conventional method assumes the expected value of the activity duration as the length of the arc in PERT network and treats the network as a deterministic one. The mean and variance of the longest path in the network are treated as the parameters of project duration. If there are more than one longest path in the network, then it takes the path with highest variance as the longest path. Thus, the conventional method assumes the maximum of means of various paths as the mean of Project duration. But by Jensen's inequality, this is only a lower bound. Thus conventional PERT method is not acceptable when the accuracy in the mean and Standard deviation of Project duration is required.

The importance in the accuracy of mean and Standard deviation of Project duration comes from the fact that these parameters are used in computing the probabilities of meeting some specified due dates and if these parameters are inaccurate, the probability statements will be very much misleading. To get a feeling of the errors in the probability statements, let us consider the following simple example:

Consider a project for which the mean and standard deviation of Project duration are found to be 22.00 and 2.00 respectively by conventional method. Then the probability of completing the project in 21 units of times is $\phi(\frac{21-22}{2.00}) = 0.5000$. Suppose the actual mean and standard deviation are 23.00 and

and 1.50 respectively. Then the probability of this event is $\phi\left(\frac{21-23}{1.5}\right) = 0.0918$. In other words, when there are only 9.18 percent of chances for completing the project in 21 units, the conventional method indicates it as 50 percent. We thus see that the statements can be highly misleading even when there are only small errors in the parameters of project duration.

The errors in the parameters of project duration become higher if there exist near critical paths in the network. For example, consider a network for which the distributions of various paths are as shown in Fig. 1.1. The critical path P_1 has parameters (m_1, σ_1). Consider a near critical path P_2 with parameters (m_2, σ_2). Now, if bad luck is generally experienced for the activities along P_2 and good luck for the activities along P_1, it is quite possible for P_2 to exceed P_1 in length — that is to become critical itself. Infact, the same may happen with other near critical paths too. Therefore, it is overly optimistic to assume (m_1, σ_1) as the parameters of Project duration as assumed in the conventional PERT.

The main idea in the present work is to reduce the errors in the parameters of Project duration by considering the effect of near critical paths. It is assumed, without loss of generality, that all the activities in the network are independent. It is also assumed that the durations of individual activities are Normally distributed. This assumption has often been made

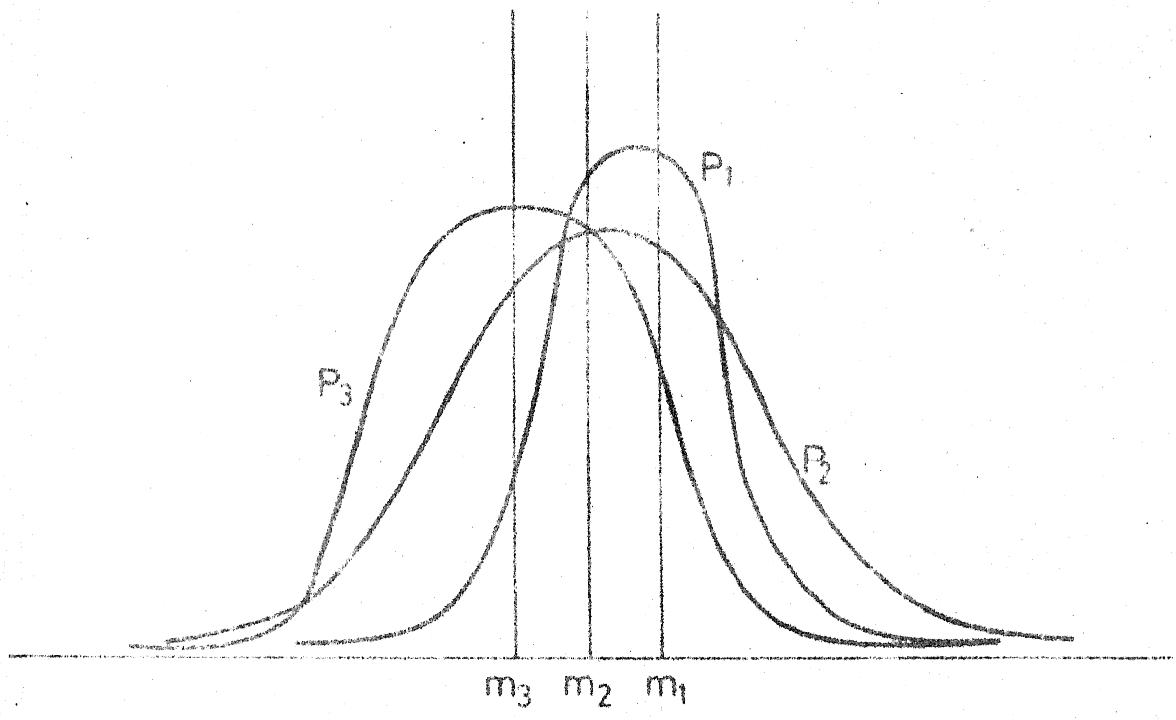

Fig. 1·1 Effect of near Critical Paths

in the literature and this can be justified by the fact that most of the networks can be reduced to a guide network, where a completely independent path becomes one activity. Then the Central limit theorem justifies the Normality assumption for the activity durations in the guide network.

1.4 FORMULATION OF THE PROBLEM:

We now give the Statistical formulation of the problem which forms the basis for our further discussions. Let us denote the various terms as follows:

t_{ij} : random variable corresponding to the duration of activity (i, j).

(m_{ij}, v_{ij}) : Parameters of t_{ij}.

P_k : Set of all the arcs on k-th path.

p_k : Random variable corresponding to P_k.

(M_k, V_k) : Parameters of p_k.

l : Random variable representing the project duration.

Now, p_k is given by,

$$p_k = \sum_{(i,j)\varepsilon P_k} t_{ij}$$

therefore,

$$M_k = E(p_k) = E\left(\sum_{(i,j)\varepsilon P_k} t_{ij}\right) = \sum_{(i,j)\varepsilon P_k} E(t_{ij})$$

$$= \sum_{(i,j)\varepsilon P_k} m_{ij}$$

Similarly,

$$V_k = \sum_{(i,j)\varepsilon P_k} v_{ij}$$

The Project duration l is given by,

$$l = \text{Max}\ \{P_1, P_2, \ldots, P_r\}.$$

We are interested in approximating the parameters of l.

1.5 OUTLINE OF THE THESIS:

A brief, chapter by chapter, outline of the thesis is given below:

In Chapter II, the existing literature related to the problem is discussed. The following methods are described in some detail.

(i) Malcolm's algorithm

(ii) Fulkerson's algorithm

(iii) Elmaghraby's algorithm

(iv) Martin's algorithm

(v) Robillard and Trahan's algorithm

(vi) Sculli's algorithm, and

(vii) Simulation approach.

Chapter III is concerned with the new approximate method developed for the problem. The intuitive ideas which underline the development of the method are first described. Then a stepwise description of the algorithm is presented. Lastly, the method is illustrated with a numerical example.

In Chapter IV, the computational investigations of the suggested method, together with the conventional method, simulation approach and Sculli's approach, are presented. The results are presented for a large number of network problems generated randomly.

A list of references and bibliography is given at the end, which is followed by program listings of all the programs used in the computational study.

CHAPTER II

LITERATURE SURVEY

2.1 INTRODUCTION:

During past several years, the problem of finding the project duration in PERT networks has received widespread interest. Most of the research was directed towards obtaining an approximate estimate because of the inherent complexity involved in evaluating its exact density function. Some researchers concentrated on finding only the mean of the project duration while others on finding both the mean and standard deviation.

In this chapter, we briefly review various approaches available in the literature. The gradual improvements in each method are discussed and their limitations are pointed out.

2.2 MALCOLM'S APPROACH:

The first approximation for the expected completion time of the PERT network was proposed by Malcolm [1], who suggested to use the expected values of the individual activities in evaluating the length of the longest path in the network. The parameters of the longest path are considered as the mean and standard deviation of the project duration. Hence the expected value of the project duration, g_n, is evaluated by

defining a function g_i recursively as follows:

$$g_1 = 0$$

$$g_i = \max_{k \varepsilon I_i} [g_k + E(t_{ki})]$$

where I_i is the set of incoming arcs at node i, and

t_{ki} is the random variable corresponding to the arc (k, i).

Thus the model is reduced to a deterministic form without incorporating the Stochastic element, variance of activity duration. That is why, in many situations, this estimate g_n is observed to be far from the real value [16].

This approach is referred to as the conventional method because of similar calculations involved as in deterministic networks.

2.3 FULKERSON'S APPROACH:

In an effort to improve the above estimate, Fulkerson [2] proposed, for discretely distributed random variables t_{ij}, the function f_i obtained recursively from

$$f_1 = 0$$

$$f_i = \sum_{k \varepsilon I_i} P(I_i) \max \{f_k + t_{ki}\}$$

where $P(I_i)$ is the joint probability distribution function of all t_{ki}, $k \varepsilon I_i$.

Thus he treated t_{ki} as random variables in evaluating f_i which allows a greater choice of path taken to have a length approximating the critical path length. Obviously, f_i is

uniformly better than g_i and in particular

$$g_n \leq f_n \leq l_n$$

where l_n is the actual expected value of project duration.

Fulkerson's approach bears straight forward generalization to the case of continuously distributed random variables. It is a well known fact that the cumulative distribution function of the maximum of a finite set of independent random variables is equal to the product of the individual cumulative functions of the variables in the set. This fact was utilized by Clingen [3] to obtain a computationally tractable form for f_i as follows:

$$f_i = b - \int_a^b \prod_{k=1}^{i-1} P_k(z-f_k)\, dz$$

where $\quad a = \max_{k \varepsilon I_i} (f_k + a_{ki})$

$\quad b = \max_{k \varepsilon I_i} (f_k + b_{ki})$

and $\quad P_k(z-f_k) = \int_{a_{ki}}^{z-f_k} p_k(x)\, dx$

where, $p_k(x)$ = density function of t_{ki}

a_{ki} = minimum possible value for t_{ki}, and

b_{ki} = maximum possible value for t_{ki}.

The limitation of the Fulkerson's approach is obvious from the expression for $P_k(z-f_k)$. If the integral of the

density function of the distribution of any activity is difficult to evaluate as in the case of Normal distributions, then this approach cannot be used. Also, this approach assumes all f_k, $k \in I_i$, as independent. But it is easy to see that they may be dependent because of the common arcs in various sub-paths reaching to nodes $k \in I_i$.

2.4 ELMAGHRABY'S METHOD:

Elmaghraby [4] extended the Fulkerson's approach in order to obtain still better approximation. His approach is based on the following simple observation: if all the arcs in a directed acyclic network are reversed, the average project duration remains unchanged. However, intermediate values of f_i, $1 < i \leq N$, donot necessarily remain the same. Consequently, f_i can only be improved if we take the maximum of the two values obtained from 'as given' and 'reversed' subnetwork.

Let μ_i be the estimate using Fulkerson's approach in the reversed subnetwork. Let s_i and u_i be defined recursively as

$$s_1 = u_1 = 0$$

$$u_i = \sum_{k \varepsilon I_i} P(I_i) \max \{s_k + t_{ki}\}$$

$$s_i = \max (u_i, \mu_i).$$

s_n is the estimate of l_n and obviously

$$f_n \leq s_n \leq l_n.$$

2.5 MARTIN'S APPROACH:

Martin [17] presented a computational method for the evaluation of the density function of the project duration in a PERT network under the assumption that the arc duration density functions are polynomials. This approach consists of mainly two steps as follows:

The first step is to transform any directed acyclic network into a series parallel network. Two subnetworks are said to be in series if the Sink of the first one is the Source of the second one. They are in parallel if both have same Source and Sink. This step is performed by depicting the PERT network as a tree where some nodes and arcs are duplicated as necessary.

The second step is to reduce systematically a series parallel network to a single equivalent arc with a density function that characterizes the project duration.

Martin developed algorithms for performing the above two steps. Unfortunately, the algorithms require a great deal of calculations even with the assumption of polynomial density functions for activity durations.

2.6 ROBILLARD AND TRAHAN'S APPROACH:

Robillard and Trahan [6] generalized the Fulkerson's estimate by extending the concept of I_i to include all the subpaths in G terminating at node i. Let P_i be the set of

subpaths terminating at node i and t_p be a random variable associated with a subpath p representing the total duration of the set of activities in p. Then

$$m_1 = 0$$

$$m_i = \max_{p \varepsilon P_i} \{m_k + t_p\}$$

where k is the beginning node of p.

It is obvious that $f_n \leq m_n \leq l_n$. But, in general, it is difficult to evaluate m_i because we have to handle many t_p simultaneously. So they proposed two modifications in the above approach which can deal only with arcs. Both these modifications require to reduce the subnetwork till node i to a parallel network as done by Martin.

The first modification is based on the observation that if we permute the arcs along a path in the reduced PERT network, the average duration of the partial project remains unchanged. However, if we calculate f_i for various permutations, it does not necessarily remain the same. If the maximum value of f_i over all the relevant permutations is chosen, it can only be better than f_i. Say this estimate as n_i^1. Therefore,

$$f_i \leq n_i^1 \leq m_i \leq l_i.$$

The second modification is to reverse all the arcs in the reduced network, as done by Elmaghraby, using n_i^1 instead of

f_i. If n_i^2 is the bound obtained by this procedure,

$$f_i \leq s_i \leq n_i^2 \leq l_i .$$

The above approaches require a lot of time for converting the subnetworks into parallel networks and for checking various permutations of arcs.

Robillard and Trahan [7] have also suggested another method for evaluating the distribution function of project duration using the approximation proposed by Davies [18] in calculating the inverse characteristic function.

2.7 SCULLI'S METHOD:

As it is already discussed, Fulkerson's approach cannot be used when the distributions of activity durations are Normal. Sculli [8] developed a method to overcome this limitation of Fulkerson's approach. He assumed that the durations of individual activities are Normally distributed and used Clark's approximate formulae for finding $f_i = \max_{k \varepsilon I_i} \{f_k + t_{ki}\}$.

It is assumed here also, as in Fulkerson's approach, that all f_k, $k \varepsilon I_i$ are independent. As a result, the errors in f_i, particularly in dense networks, are more because of large number of dependent paths terminating at nodes $k \varepsilon I_i$.

Many other methods based on similar ideas have been proposed. Lindsey [5] developed an estimate for the project duration based on a model approximating the actual network. This estimate may fall on either side of actual value l_n. Its calculation is more complicated and it did not improve the Fulkerson's estimate significantly except for nodes nearer to Source node. Kleindorfer [20] tried to find the distribution functions that bound the activity starting and completion. In practice, this approach concerns only PERT networks where the distribution functions associated with the activities are discrete.

2.8 SIMULATION APPROACH:

A very close approximations to the mean and standard deviation of project duration in PERT networks can be obtained by using Monte Carlo Simulation [19]. Given the distributions of activity durations, the network can be simulated by generating the durations of activities randomly and finding the project duration using conventional method. In each run, we get a value for the project duration and the mean and standard deviation of all the values obtained for different runs gives the parameters of the project duration.

The advantage of this approach is that it considers all the possible variabilities in activity times if the network is simulated for sufficient number of runs. So this method can be taken as an accurate method for finding the parameters of

project duration. The major drawback in this method is that it requires a lot of time for simulating even a reasonably large network.

CHAPTER III

PATH ENUMERATION ALGORITHM

3.1 INTRODUCTION:

In this chapter, a new approximate method for finding the mean and standard deviation of project duration in PERT networks is developed. The method is called Path Enumeration Algorithm (PEA) as it is based on the enumeration of k-th longest paths in the network. We first describe the intuitive ideas which led to its development. A stepwise description of the algorithm is then presented. A numerical example is also given to illustrate the algorithm.

3.2 DEVELOPMENT OF PATH ENUMERATION ALGORITHM:

As already discussed in Section 1.4, the actual project duration in PERT networks is given by $l = \underset{i}{\mathrm{Max}} \{p_i\}$ where p_i is the random variable corresponding to the i-th path in the network which is assumed to be normally distributed with parameters (M_i, V_i). The random variables p_i's are dependent because of the common areas in different paths in most of the networks. There is no method available to find the parameters of maximum of a set of normal random variables even for independent variables. Dependency among the paths makes the problem more difficult. Also the number of paths in a network may be extremely large even for networks of reasonable size. Hence it is quite unlikely to

enumerate all the paths in the network. It is, infact, not necessary to enumerate all the paths because only a set of paths generally dominate all other paths in the network. In the path enumeration algorithm, we enumerate this set of paths and heuristically try to find their maximum considering the dependency among themselves.

It is easy to see that the paths with higher value of expected path length tend to dominate those paths with lower values of expected path lengths. We therefore enumerate the paths in the non-increasing order of their expected lengths and consider paths until further enumeration changes the parameters of maximum of such paths by negligible amount.

The enumeration of paths in the required order can be efficiently done by Dreyfus method [14]. A procedure for finding the parameters of maximum of the random variables corresponding to the paths, which are Normally distributed is yet to be discussed. One obvious method is to use Clark's formulae recursively. Simulation technique can also be used for this problem. In each simulation run, we can find the maximum value of the variates generated randomly with the parameters of the random variables and then the mean and standard deviation of all such values for various runs gives the parameters of maximum of the random variables. Both the above methods were programmed and the values of the parameters were found to be comparable.

Hence Clark's method, which takes lesser time than simulation is selected to find the parameters of maximum of random variables.

Thus the procedure for finding the parameters of project duration is to enumerate k-th longest path in the network and to find the parameters of maximum of this path with the previous paths using Clark's formulae and repeat these two steps until the enumeration of a path does not change the parameters significantly.

The results from the above procedure were found to be substantially different from those with simulation approach. These deviations are anticipated since Clark's formulae treats the random variables as independent in finding the parameters of their maximum. But these random variables, which correspond to different paths in the network, may be dependent as there may be some common arcs. Until we consider the dependency among the paths in an effective way, accurate results may not be obtained. We now describe a method to take care of dependency among the paths.

Any two paths, P_1 and P_2, in a network may have some common arcs and some uncommon arcs. If the sets of uncommon arcs are separated, then the two paths may be as shown in Fig.3.1. To consider dependency between P_1 and P_2, the parameters of the maximum of only the uncommon portions can be found and added

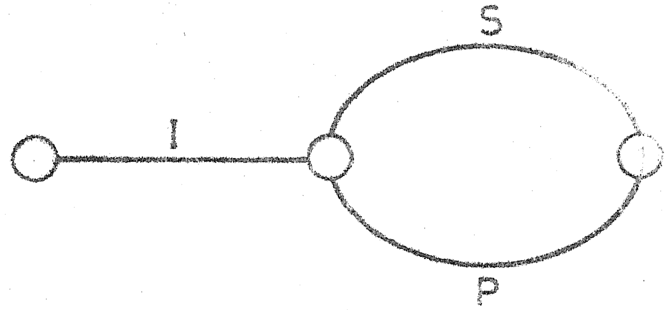

Fig. 3·1 Showing the seperated common and uncommon portions

to the parameters of common portion. We describe this in a more formal manner. Let,

$$S = \text{Set of arcs in } P_1 \text{ but not in } P_2 = P_1 - P_1 \cap P_2$$

$$P = \text{Set of arcs in } P_2 \text{ but not in } P_1 = P_2 - P_1 \cap P_2$$

and

$$I = \text{Set of common arcs in } P_1 \text{ and } P_2 = P_1 \cap P_2.$$

The arcs in the set I can be easily found by labelling all the arcs on the first path P_1. The parameters of the common portion can then be calculated as,

$$C_3 = \sum_{(i,j)\varepsilon I} m_{ij}$$

and,

$$D_3 = \sum_{(i,j)\varepsilon I} v_{ij}.$$

If $\bar{\bar{C}}$ and $\bar{\bar{D}}$ denote the random variables corresponding to the uncommon portions, their parameters are given by,

$$(C_1, D_1) = (M_1 - C_3, V_1 - D_3)$$

and

$$(C_2, D_2) = (M_2 - C_3, V_2 - D_3), \text{ respectively.}$$

Now the procedure for finding the parameters of maximum of two dependent random variables corresponding to the paths P_1 and P_2 is to find the parameters of maximum of the independent random variables $\bar{\bar{C}}$ and $\bar{\bar{D}}$ using Clark's formulae and add them to (C_3, D_3).

This procedure is to be extended for finding the maximum of a set of dependent random variables in order to make it usable

in PERT networks. The difficulty in extending it is that a new path may have different arcs in common with different paths enumerated and so it is difficult to decide which arcs are to be included in the set I. An analogy with the procedure for two paths in the network suggests to include the arcs which are common to the new path and to any of its previous paths. So all the arcs on the previous paths are to be included first in the set S and then the arcs in set I can be easily identified by labelling all the arcs in the set S.

The effect of the arcs in the set I, which are causing for the dependency, is now to be considered. The enumeration of a new path P changes the parameters of the maximum of all the previous paths (M, V_1) to certain values say (M_2, V_2) with $M_2 > M_1$ and $V_2 < V_1$. Thus the affect of the arcs in P is to increase M_1 to M_2 and reduce V_1 to V_2. In order to consider this effect in the further enumeration, the mean of these arcs is to be reduced by a factor of M_1/M_2 and the variance to be increased by a factor of V_1/V_2. Let us call the calculated means and variances using these factors as <u>effective means</u>, $\bar{m}_{ij}$, and <u>effective variances</u>, $\bar{v}_{ij}$ respectively. Therefore,

$$\bar{m}_{ij} = \bar{m}_{ij} \times M_1/M_2 \qquad \forall\ (i,j)\ \varepsilon\ P$$

$$\bar{v}_{ij} = \bar{v}_{ij} \times V_1/V_2 \qquad \forall\ (i,j)\ \varepsilon\ P.$$

The above formulae require the initialisation of $\bar{m}_{ij}$ and $\bar{v}_{ij}$ for each arc in the network as equal to its actual mean and actual variance.

The effective means and effective variances are now to be used in finding the parameters of the common and uncommon portions. Thus,

$$C_3 = \sum_{(i,j)\varepsilon I} \tilde{m}_{ij} \qquad \text{and} \qquad D_3 = \sum_{(i,j)\varepsilon I} \tilde{v}_{ij}$$

$$C_1 = M_1 - C_3 \qquad \text{and} \qquad D_1 = V_1 - D_3$$

In the above expression for D_1, the term V_1 signifies the variance of all the arcs in the set S and D_3 signifies the variance of the labelled arcs in the path P. V_1 is calculated as the variance of maximum of all the previous paths but D_3 is calculated, in a different way, as the sum of $\tilde{v}_{ij}$ for the labelled arcs. Thus D_3 is not in the same scale of V_1. To bring it into the same scale of V_1, we use a scaling factor of V_1/V_s where V_s is the sum of $\tilde{v}_{ij}$ for all the arcs in S. Thus,

$$D_3 = D_3 \times (V_1/V_s) ,$$

where, $V_s = \sum_{(i,j)\varepsilon S} \tilde{v}_{ij}$

the scaling factor V_1/V_s is less than or equal to 1.

We thus have a heuristic procedure for finding the parameters of maximum of random variables, taking care of dependency among the paths. This procedure can be repeated, after the enumeration of each path, until the difference between the successive values or the parameters of project duration is less than a small number, ε, for a number of times.

3.3 STATEMENT OF THE ALGORITHM:

Path Enumeration Algorithm can be described stepwise as follows:

Step 1: Enumerate first longest path P using Dijkstra's algorithm. Let,

$$M_1 = \sum_{(i,j)\varepsilon P} m_{ij} \quad \text{and} \quad V_1 = \sum_{(i,j)\varepsilon P} v_{ij}$$

Set,
$$L_{ij} = 1 \quad \forall\ (i,j)\ \varepsilon\ P$$
$$\bar{m}_{ij} = m_{ij} \quad \forall\ (i,j)\ \varepsilon\ A$$
$$\bar{v}_{ij} = v_{ij} \quad \forall\ (i,j)\ \varepsilon\ A$$
$$V_s = V_1\ ,\quad S = P \quad \text{and} \quad k = 2.$$

Step 2: Enumerate the k-th longest path P using Dreyfus method. Let,

$$X = \sum_{(i,j)\varepsilon P} m_{ij}$$

If $(X = \infty)$, then go to Step 9, else go to Step 3.

Step 3: Define, $I = S \cap P$

Set, $M_2 = M_1$ and $V_2 = V_1$

$$\bar{M} = \sum_{(i,j)\varepsilon P} \bar{m}_{ij} \quad \text{and} \quad \bar{V} = \sum_{(i,j)\varepsilon P} \bar{v}_{ij}$$

Define the parameters of the common portion I as follows:

$$C_3 = \sum_{(i,j)\varepsilon I} \bar{m}_{ij} \quad \text{and}$$

$$D_3 = \sum_{(i,j)\varepsilon I} \bar{v}_{ij} \times V_1/V_s$$

Define the parameters of uncommon portions as follows:

$$C_1 = M_1 - C_3, \quad D_1 = V_1 - D_3$$
$$C_2 = \bar{M} - C_3, \quad D_2 = \bar{V} - D_3$$

If $C_1 = 0$ then go to Step 8, else go to Step 4.

Step 4: Find the parameters (C_e, D_e) of the maximum of uncommon portions as follows:

Set $a = D_1^2 + D_2^2$ and $\alpha = (C_1 - C_2)/a$

$$C_e = C_1\,\phi(\alpha) + C_2\,\phi(-\alpha) + a\,\psi(\alpha)$$
$$D_e = [(C_1^2 + D_1)\,\phi(\alpha) + (C_2^2 + D_2)\,\phi(-\alpha) + (C_1 + C_2)\,a\,\psi(\alpha)] - C_e^2.$$

Set $M_1 = C_e + C_3$ and $V_1 = D_e + D_3$

If $(M_1 - M_2 > \varepsilon)$ then go to Step 6, else go to Step 5.

Step 5: Set count = count + 1

If (Count > 5) then go to Step 9, else go to Step 7.

Step 6: Set count = 0.

Step 7: Update $\bar{m}_{ij}$ and $\bar{v}_{ij}$ as follows:

$$\bar{m}_{ij} = \bar{m}_{ij} \times M_1/M_2 \qquad \forall\,(i,j) \in P$$
$$\bar{v}_{ij} = \bar{v}_{ij} \times V_1/V_2 \qquad \forall\,(i,j) \in P$$

Update the labels and V_s as follows:

$$L_{ij} = 1 \qquad \forall\,(i,j) \in P - P \cap S.$$
$$V_s = V_s + \sum_{(i,j)\in P - P\cap S} \bar{v}_{ij}$$

Set $S = S \cup P$

Step 8: Set $k = k + 1$

If ($k \geq$ Limit) then go to Step 9, else go to Step 2.

Step 9: M_1 and V_1 are the mean and variance of the project duration. STOP.

3.4 NUMERICAL EXAMPLE:

We now solve a numerical example to illustrate the various steps of the algorithm. Since the main obstacle in computing the parameters of project duration stems from the possibility of non-critical paths becoming critical, the network shown in Fig. 3.2 is analysed where all the nine possible paths are expected to be critical. In this network, it is assumed that the individual activities have a Normal distribution with variance equal to 20 percent of their mean. The mean values are mentioned over the arcs in the network.

The sets P and S at each iteration and the parameters of project duration considering the affect of all the previous paths upto that iteration are shown in Table 3.1. The iteration number indicates the number of longest paths that have been enumerated till that iteration. For this problem, the value of ε is fixed as 0.2 percent of the mean of first longest path and the number for which the difference between successive values is less than ε is fixed as 5. It is terminated after evaluating all the paths in the network because the condition with ε is not satisfied. The parameters obtained at the terminating point

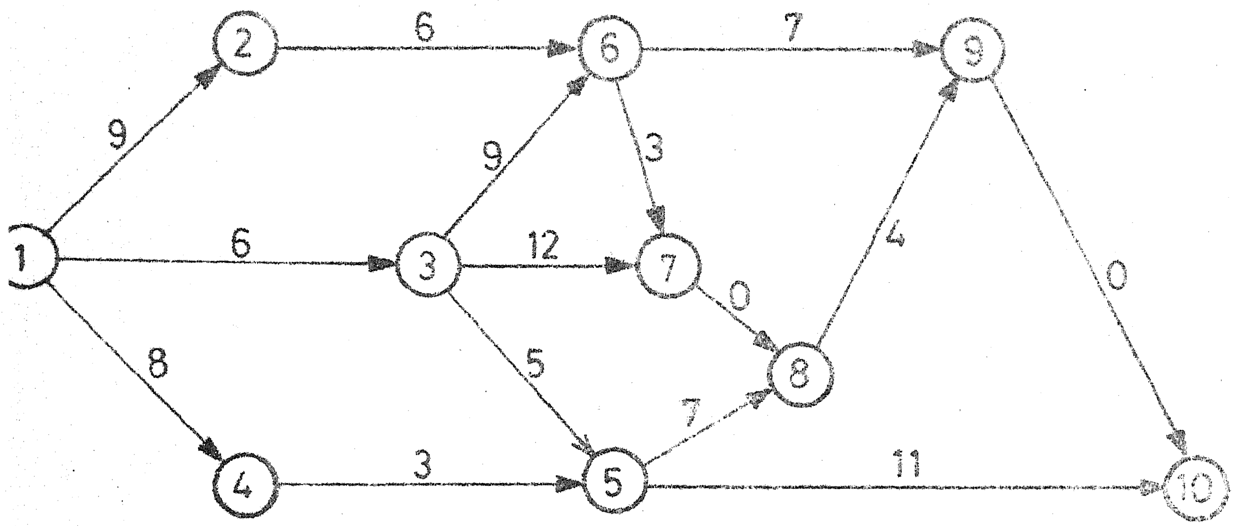

Fig. 3·2 Numerical Example Network

TABLE 3.1: Solution of the Numerical example.

Iteration No.	Activities in P	Activities in S	Parameters of Project Duration: Mean	Variance
1.	{2, 5, 10}	{2, 5, 10}	22.00	4.40
2.	{1,4,12,15}	{1,2,4,5,10,12,15}	23.18	3.00
3.	{2,5,9,14,15}	{1,2,4,5,9,10,12,14,15}	23.67	2.62
4.	{2,7,13,14,15}	{1,2,4,5,7,9,10,12,13,14,15}	23.99	2.37
5.	{1,4,11,13,14,15}	{1,2,4,5,7,9,10,11,12,13,14,15}	24.16	2.34
6.	{2,6,11,13,14,15}	{1,2,4,5,6,7,9,10,11,12,13,14,15}	24.36	2.19
7.	{3,8,9,14,15}	{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15}	24.54	2.05
8.	{2,6,12,15}	{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15}	24.54	2.05
9.	{3,8,10}	{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15}	24.54	2.05

are Mean = 24.54 and Standard deviation = 1.43. The parameters obtained by Simulation approach for a total of 10000 runs are 24.67 and 1.47 respectively.

CHAPTER IV

COMPUTATIONAL INVESTIGATIONS

4.1 INTRODUCTION:

In this chapter, we present the computational performance of Path Enumeration Algorithm developed in Chapter III on randomly generated network problems. The results obtained by simulation approach are considered as correct results as no exact method is available.

The computational experiments require a large number of randomly generated networks. The details of the network requirements and the type of network problems considered for computational study are discussed in Section 2. Section 3 covers the explanation of various algorithms used as subroutines while programming Path enumeration algorithm and other methods. The computational considerations such as computational times, accuracy, storage requirements etc. are also given, for analysing the algorithms, in this section. In Section 4, the computational experiments with various parameters are explained and the results of Path enumeration algorithm are compared against conventional, simulation and Sculli's methods. The advantages and limitations of Path enumeration algorithm are also given in this section.

4.2 TOPOLOGICAL STRUCTURES OF RANDOM NETWORKS:

It is mentioned in Section 1.2.2 that the PERT networks can be only directed acyclic networks. Keeping this point in view, the following three random network generators are used in the computational study. These generate topologically sorted networks.

(i) Network Generator 1 (NG 1):

NG 1 takes the network dimensions, width (w) and length (L), as input and generates a sparse networks. An example network generated from NG 1 with a width of 3 and a length of 5 is shown in Fig. 4.1. There are 17 nodes and 36 arcs in this network. Infact, one can derive the general formulae for finding the number of nodes, n, and the number of arcs, m, in the network as follows:

$$n = W.L + 2$$

$$m = L\,(3W - 2) + 1$$

(ii) Network Generator 2 (NG 2):

NG 2 takes n as the input and generates a complete acyclic network. Therefore the networks generated by NG 2 are dense with $m = \binom{n}{2}$. An example network generated by NG 2 with $n = 6$ is shown in Fig. 4.2.

(iii) Network Generator 3 (NG 3):

In the previous network generators, there was no control over the number of arcs. In NG 1, for a given width and length,

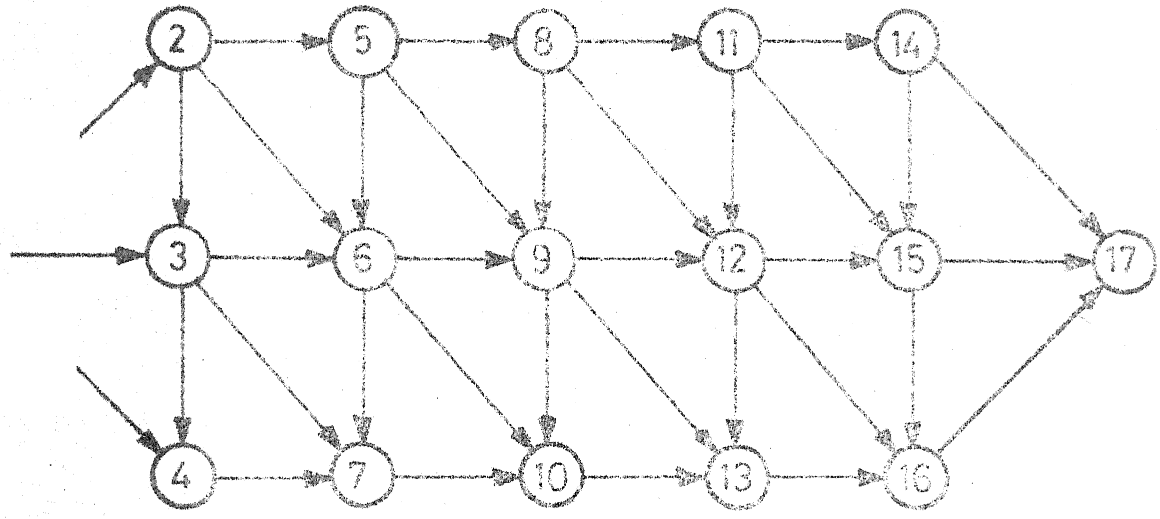

Fig. 4·1 Example Network for NG 1
Width = 3 ; Length = 5

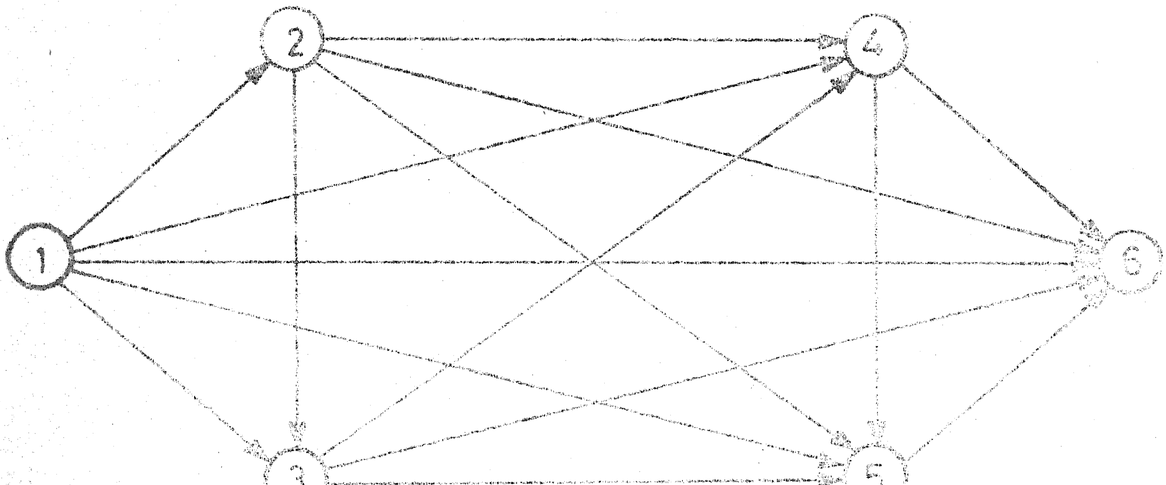

Fig. 4·2 Example Network for NG 2

both n and m are fixed. Similarly in NG 2, for a given n, m is fixed. In NG 3, we have control over both the parameters n and m. The number of arcs can also be specified in addition to the width and length of the network. Then NG 3 first generates a skeleton network with the given width and length, and then it adds arcs between two randomly generated nodes until the total number of arcs becomes the specified m. An example for the skeleton network is shown in Fig. 4.3 for a width of 2 and length of 5. The final network, generated by NG 3, after augmenting the additional arcs for a given m = 19 is shown in Fig. 4.4.

The computational study has been made with a large number of problems generated by these three network generates. We feel that these three generators are sufficient for our study and the results thus obtained will hold good for real-life network problems.

In all these network generators, the networks are first generated in Arc list representation and then converted into Adjacency array representation. In Arc list representation, the head and tail nodes of each arc, together with the arc number, are maintained. Therefore if there are m arcs in the network, the total storage requirement is 3 m. If all the arcs incident from or incident to a node are to be retrieved, the whole list of arcs is to be scanned. In Adjacency array

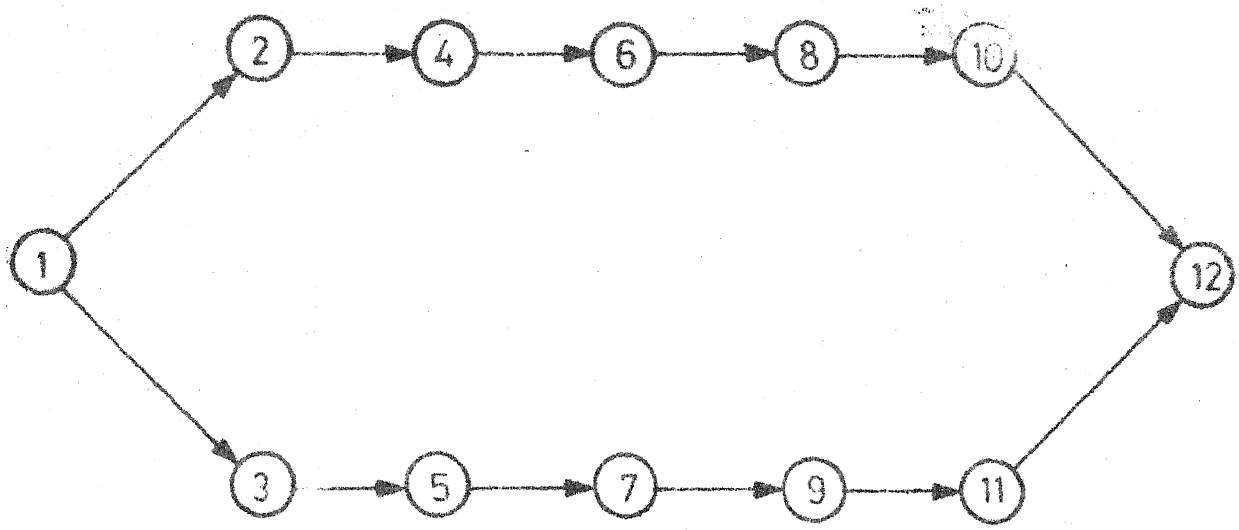

Fig. 4·3 Skeleton Network for NG 3
Width = 4 ; Length = 6

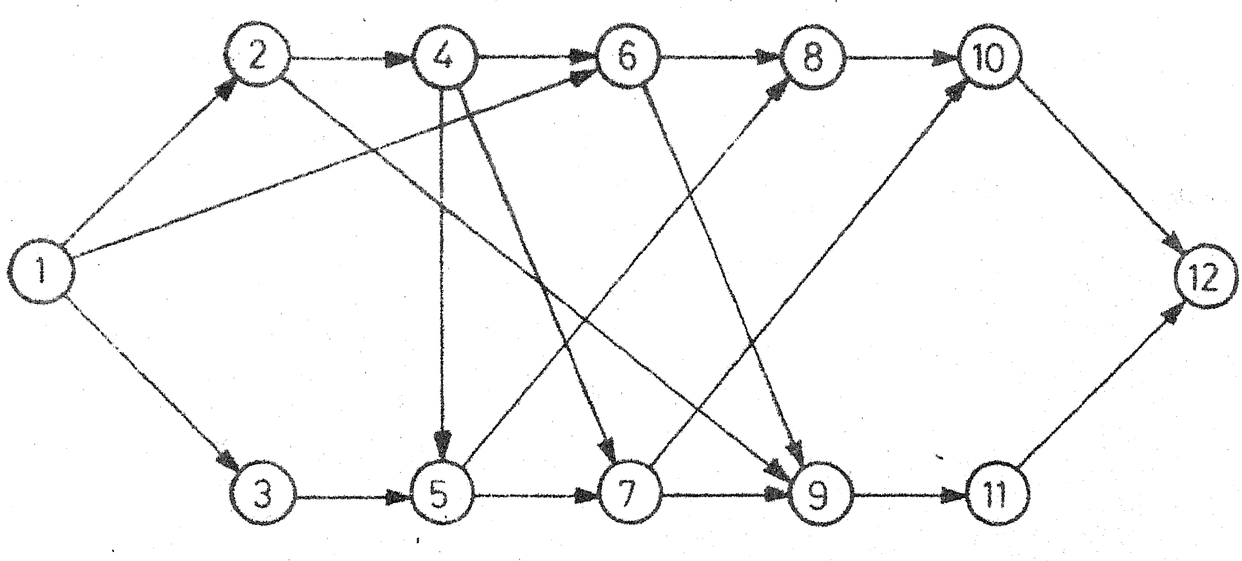

Fig. 4·4 Network after augmenting the additional area
m = 17

representation, two arrays POINT and LIST are maintained. LIST contains all the arc numbers incidenting from all the nodes with the nodes arranged in sequential order. The position at which the arcs incident from a node starts, is stored in POINT array. The difference between POINT (i+1) and POINT (i) gives the number of arcs emanating from the node i. Thus, by knowing the starting position in the LIST and the number of arcs, the arcs incident from that node are retrieved. Similarly the arcs incident to a node can be retrieved easily by maintaining two arrays RPOINT and RLIST.

4.3 SOME COMPUTATIONAL CONSIDERATIONS:

(i) Generation of Normal random variates:

There are three well known approaches for generating the Normal variates randomly, which we discuss briefly. Let U_i be a Uniformly distributed random variate defined over the interval (0,1) and X_i be a standard Normal variate.

The first approach, discussed in Fishman [10], uses the following formulae for generating X_i.

$$X_1 = (-2 \log_e U_1)^{1/2} \cos 2\pi U_2$$

$$X_2 = (-2 \log_e U_1)^{1/2} \sin 2\pi U_2.$$

The second approach, given in Naylor [11] uses

$$X = \left(\frac{12}{k} \right)^{1/2} \left(\sum_{i=1}^{k} U_i - \frac{k}{2} \right).$$

This is referred to as "central limit approach". The number k, the number of U_i's to be generated, must be chosen depending upon the required accuracy and computational efficiency. A high value of k is desirable, say around 24, is desirable for accurate results. But for computational simplicity, one may use k = 12.

The third method, discussed in Law and Kelton [12] is as follows:

Step 0: Generate U_1 and U_2

Say $V_1 = 2U_1 - 1$ and $V_2 = 2U_2 - 1$

Set $W = V_1^2 + V_2^2$

If $(W > 1)$ then go to Step 0, else go to Step 1.

Step 1: Put $Y = (-2 \log_e W)/W$

$X_1 = V_1 Y$ and $X_2 = V_2 Y$.

The above three methods were programmed and histograms were plotted by generating 1000 random variates for a given parameters of Normal random variable. The shapes of the histograms obtained with the third method were observed to be very much similar to the Normal distributions of input parameter. Hence, the third method was found to be most appropriate for generating Normal random variates and used in the computational study.

(ii) No. of Simulation Runs in Simulation Approach:

For obtaining the mean and standard deviation of project duration using simulation approach, the network was simulated for a total of 1000 runs. It was observed that the values of the parameters get stabilized after performing 1000 simulated trials on the network.

(iii) Generation of the Parameters of Activity Durations:

After generating the network structures using the network generators, the means and variances of durations of all the activities in the network are to be generated. These are generated in three ranges individually. The first range considered is (5, 10), which is selected as narrow in order to clearly bring out the effects of near critical paths in the network. The next range is chosen as (1, 100) in order to make the situation closer to more practical situations. The third range considered is (5, 50).

(iv) The Values of ε and Limit in Path Enumeration Algorithm:

In the Path enumeration algorithm, a small value of ε is used to check for the stabilization of the parameters of project duration. The stabilization of these parameters depends upon the number of paths enumerated, which inturn depends upon many factors such as network size and structure, parameters of individual activities etc. Thus the value of ε is correlated to a given instance of the problem. In an effort to relate ε

with the problem, by a constant observation of several problems, it was found that the parameters of project duration would not change once the difference between successive values becomes nearly 0.2 percent of the mean of the first longest path for four or five times. Thus the value of ε was fixed as 0.2 percent of the mean of the first longest path in the network and it was checked, for termination, whether the difference between successive parameters is less than ε for five consecutive times. Also, the parameters were observed to stabilize after the enumeration of only a few paths in the network and in most of the cases, before enumerating 40 paths. Thus the limit for the number of paths to be enumerated was put as 40.

4.4 COMPUTATIONAL RESULTS:

A computer program was written in FORTRAN-IV for the Path enumeration algorithm. The program stores the network in adjacency array representation. The conventional method, simulation method and Sculli's method were also programmed. The listings of the programs are included in the Appendix. These programs were debugged and tested on DEC-1090 multiprogramming, time-sharing computer system.

The main emphasis in the computational study was laid on (i) the accuracy of the results, and (ii) the computational times taken by various methods. While solving the Path enumeration algorithm, the number of paths enumerated for reaching

the results is also emphasized. Since these factors depend upon the network size and its structure, a large number of problems generated with the three kinds of network generators NG 1, NG 2 and NG 3 were used in the study. The emphasized factors also depend upon the means and variances of individual activities in the network. Hence three ranges, as discussed earlier, were used for generating these parameters. If (k1, k2) is the range used, the parameters were generated as follows:

$$m_{ij} = \text{RAND } (k1, k2) \quad \forall \, (i,j) \in A$$
$$v_{ij} = \text{RAND } (k1, k2) \quad \forall \, (i,j) \in A$$

where RAND (k1, k2) is a function that generates a uniformly distributed random number in the interval (k1, k2).

The accuracy in the parameters obtained by each method is measured by taking the percentage deviations from the simulated values. Since for large networks, with a higher value of project duration, the percentage deviations are not appropriate for measuring the accuracy, the actual values obtained by each method are also shown for each problem.

To generate the network problems by NG 1, the length of the network was varied from 5 to 30 and width from 5 to 8. All these problems were first solved by fixing the range for the parameters of activity durations as (5, 10). Table 4.1 shows the results of these problems for various methods. The same problems were then solved for other ranges (5, 50) and (1,100). These results are presented in Tables 4.2 and 4.3 respectively.

In the network generator NG 2, the parameter n was varied from 5 to 60 and the problems generated were solved for all the three ranges (5, 10), (5, 50) and (1, 100). These results are presented in Tables 4.4 to 4.6 respectively.

A number of problems were generated by NG 3 also by varying the width from 3 to 10 and length from 5 to 30. The number of arcs were chosen reasonably to relate the problems with real life problems. Tables 4.7 to 4.9 present the results for these problems.

To have a clear picture of the errors in the parameters with various methods, the percentage deviations are plotted for each problem. Fig. 4.5 shows these deviations for the first six problems shown in Table 4.1. The errors in mean are shown in Fig. 4.5(A) and the errors in standard deviation are shown in Fig. 4.5(B). Similar kind of plots are shown in Fig. 4.6 and Fig. 4.7 for the network problems generated by NG 2 and NG 3 respectively.

The execution times for various methods are noted as follows (the execution time is the CPU time exclusive of input and output times): A network is generated with given dimensions. Five different problems, for the same network structure, were generated by varying the seed from 0 to 40 in steps of 10. These five problems were then solved and their average execution time was noted. Tables 4.10 to 4.12 shows these times for

the network problems generated by NG 1, NG 2 and NG 3 respectively. The average execution times of both conventional method and Sculli's method are always lesser than those of Path enumeration algorithm. The results shown in Table 4.12 (for NG 3) are plotted and shown in Fig. 4.8.

The computational experiments prove that the conventional method, as anticipated, always under-estimates the mean and over-estimate the standard deviation of project duration. The errors are more when the parameters of activity durations are in the range (5, 10). This is due to the effect of many near critical paths in the problems generated with that range. Sculli's method always over-estimate the mean and under-estimate the standard deviation. This was also expected because the dependency among paths was not considered in this method. The errors are more with the dense networks generated by NG 2 because of more number of dependent paths terminating at each node.

Path enumeration algorithm performs as a compromise between the above two methods. The deviations in the parameters may fall on either side of simulated values. The reason is as follows: The algorithm starts with the results of conventional method and improves them gradually by taking the effect of near critical paths, until the successive values of the parameters differ by a negligible amount. In this process, the mean of the project duration may exceed the simulated value by a small amount.

The improved results with the Path enumeration algorithm are obtained at a cost of extra computing effort. However, the total effort is still well below the effort required in simulation method. The algorithm requires a higher computer storage when compared with the storage requirements of other methods. This is due to the reason that Dreyfus method requires to store all the previous paths when it evaluates k-th longest path. This limits us to solve still larger size problems. The computer program occupies mk + 3nk + 8m + 3n words of central memory.

4.5 CONCLUSIONS:

In this thesis work, a new approximate method for determining the completion times of PERT networks is developed. Computational analysis over a large number of randomly generated network problems shows that the method will prove of assistance when the accuracy in the parameters of project duration is important. The method can solve practically large problems in reasonable amount of computer time. Thus it may be concluded that the method is 'appreciably-acceptable' for estimating the project duration in PERT networks.

Though it is assumed that the activity durations are Normally distributed, the method can be used for any kind of distributions, even with different distributions for different activities, because the paths will always be Normally distributed.

Sl. No.	Network Dimensions				Simulation	Conventional	Sculli	PEA	Percent Deviations from Simulated Values		
	W	L	n	m					Conventional	Sculli	PEA
1.	5	5	27	66	91.15 6.79	84.00 8.12	94.69 4.78	91.85 6.00	7.84 19.59	3.89 29.70	0.78 -11.67
2.	5	15	77	196	189.50 9.48	170.00 11.83	197.59 5.42	182.66 9.31	10.29 24.86	4.27 42.80	- 3.61 - 1.80
3.	5	20	102	261	230.16 9.83	205.00 13.45	240.20 5.22	223.64 9.61	10.93 36.89	4.36 46.84	- 2.83 - 2.20
4.	5	25	127	326	273.56 10.84	248.00 15.20	285.53 5.04	266.94 10.98	9.34 40.16	4.38 53.48	- 2.42 1.26
5.	5	30	152	391	312.81 12.01	286.00 16.49	328.26 5.17	305.93 12.23	8.57 37.34	4.94 56.92	- 2.20 1.84
6.	8	5	42	111	121.62 7.57	111.00 10.00	126.45 4.81	119.55 7.90	8.73 32.15	4.14 36.41	- 1.70 4.37
7.	8	15	122	331	219.68 8.89	196.00 13.15	226.16 4.88	211.24 9.35	10.78 47.88	2.95 45.14	- 3.84 5.07
8.	8	20	162	441	262.83 9.69	232.00 14.04	271.79 5.03	248.79 10.50	11.73 44.90	3.41 48.08	- 5.34 8.41
9.	8	25	202	551	310.42 10.87	276.00 15.46	320.32 5.17	296.00 11.36	11.09 42.26	3.19 52.44	- 4.64 4.55
10.	8	30	242	661	357.60 11.86	321.00 16.58	369.07 5.29	340.67 12.65	10.24 39.87	3.21 55.35	- 4.74 6.67

TABLE 4.2: Results from NG_1(5, 50).

Sl. No.	Network Dimensions W	L	n	m	Simulation	Conventional	Sculli	PEA	Percent Deviations from Simulated Values Conventional	Sculli	PEA
1.	5	5	27	66	343.09 13.41	340.00 14.28	347.68 11.23	343.46 13.46	0.90 6.52	1.34 16.24	0.11 0.36
2.	5	15	77	196	727.70 19.59	721.00 21.52	737.67 13.21	725.40 19.72	0.92 9.87	1.37 32.53	- 0.32 0.69
3.	5	20	102	261	854.73 21.06	844.00 25.20	865.73 14.48	848.74 23.56	1.26 19.67	1.29 31.23	- 0.70 11.88
4.	5	25	127	326	1011.70 22.79	983.00 28.12	1028.53 13.20	1014.10 21.14	2.84 23.40	1.66 42.07	0.24 - 7.26
5.	5	30	152	391	1156.00 24.80	1127.00 31.50	1177.48 13.21	1160.71 23.46	2.51 27.02	1.86 46.71	0.41 - 5.39
6.	8	5	42	111	470.19 17.39	465.00 19.13	479.43 12.25	470.82 17.13	1.10 10.03	1.96 29.55	0.13 - 1.45
7.	8	15	122	331	829.22 20.94	820.00 24.86	842.12 14.88	827.14 23.07	1.11 18.74	1.56 28.81	- 0.25 10.21
8.	8	20	162	441	968.18 20.81	942.00 28.09	984.42 12.47	968.42 20.63	2.70 34.96	1.68 40.10	0.02 - 0.88
9.	8	25	202	551	1147.25 23.81	1119.00 31.03	1165.57 13.26	1144.72 24.00	2.46 30.34	1.60 44.33	- 0.22 0.78
10.	8	30	242	661	1353.97 27.74	1327.00 32.91	1372.09 15.35	1352.79 26.13	1.99 18.65	1.34 44.67	- 0.09 - 5.80

TABLE 4.2: Results from NG1 (5, 50).

Sl. No.	Network Dimensions W	L	n	m	Simulation	Conventional	Sculli	PEA	Percent Deviations from Simulated Values: Conventional	Sculli	PEA
1.	5	5	27	66	343.09 13.41	340.00 14.28	347.68 11.23	343.46 13.46	0.90 6.52	1.34 16.24	0.11 0.36
2.	5	15	77	196	727.70 19.59	721.00 21.52	737.67 13.21	725.40 19.72	0.92 9.87	1.37 32.53	- 0.32 0.69
3.	5	20	102	261	854.73 21.06	844.00 25.20	865.73 14.48	848.74 23.56	1.26 19.67	1.29 31.23	- 0.70 11.88
4.	5	25	127	326	1011.70 22.79	983.00 28.12	1028.53 13.20	1014.10 21.14	2.84 23.40	1.66 42.07	0.24 - 7.26
5.	5	30	152	391	1156.00 24.80	1127.00 31.50	1177.48 13.21	1160.71 23.46	2.51 27.02	1.86 46.71	0.41 - 5.39
6.	8	5	42	111	470.19 17.39	465.00 19.13	479.43 12.25	470.82 17.13	1.10 10.03	1.96 29.55	0.13 - 1.45
7.	8	15	122	331	829.22 20.94	820.00 24.86	842.12 14.88	827.14 23.07	1.11 18.74	1.56 28.81	- 0.25 10.21
8.	8	20	162	441	968.18 20.81	942.00 28.09	984.42 12.47	968.42 20.63	2.70 34.96	1.68 40.10	0.02 - 0.88
9.	8	25	202	551	1147.25 23.81	1119.00 31.03	1165.57 13.26	1144.72 24.00	2.46 30.34	1.60 44.33	- 0.22 0.78
10.	8	30	242	661	1353.97 27.74	1327.00 32.91	1372.09 15.35	1352.79 26.13	1.99 18.65	1.34 44.67	- 0.09 - 5.80

TABLE 4.3: Results from NG 1 (1, 100).

Sl. No.	Network Dimensions W	L	n	m	Simulation	Conventional	Sculli	PEA	Percent Deviations from Simulated Values: Conventional	Sculli	PEA
1.	5	5	27	66	653.21 17.96	651.00 18.68	658.62 15.96	653.46 18.24	0.34 4.01	0.83 11.13	0.04 1.54
2.	5	15	77	196	1385.54 27.71	1383.00 28.57	1395.23 20.45	1387.12 26.34	0.18 3.10	0.70 26.20	0.11 -4.94
3.	5	20	102	261	1610.94 31.04	1606.00 33.84	1620.45 24.88	1610.58 31.77	0.31 9.01	0.59 19.83	-0.02 2.34
4.	5	25	127	326	1893.06 31.35	1860.00 39.80	1908.67 20.94	1899.31 29.45	1.75 26.93	0.82 33.21	0.33 -6.07
5.	5	30	152	391	2157.90 33.68	2122.00 42.81	2179.14 20.77	2165.94 32.35	1.66 27.12	0.99 38.34	0.37 -3.96
6.	8	5	42	111	896.17 24.09	891.00 25.85	906.26 18.46	895.94 24.16	0.58 7.31	1.13 23.33	-0.02 0.30
7.	8	15	122	331	1577.95 31.32	1571.00 33.56	1592.60 23.56	1577.41 31.64	0.44 7.14	0.93 24.77	-0.03 1.01
8.	8	20	162	441	1816.94 30.11	1797.00 38.29	1836.70 20.45	1819.02 29.23	1.10 27.16	1.09 32.09	0.11 -2.93
9.	8	25	202	551	2154.10 35.38	2136.00 42.37	2174.82 22.24	2146.07 38.66	0.84 19.75	0.96 37.14	-0.37 9.28
10.	8	30	242	661	2557.00 41.21	2539.00 44.80	2574.11 27.02	2550.13 40.96	0.70 8.72	0.67 34.42	-0.27 -0.61

TABLE 4.4: Results from NG 2 (5, 10)

Sl. No.	n	m	Simulation	Conventional	Sculli	PEA	Percent Deviations from Simulated Values: Conventional	Sculli	PEA
1.	5	10	29.49	29.00	29.92	29.50	1.68	1.44	0.02
			4.69	5.10	4.23	4.64	8.80	9.81	-0.96
2.	15	105	102.23	99.00	109.09	104.06	3.16	6.70	1.79
			8.38	10.05	4.67	8.55	19.89	44.32	1.95
3.	21	210	150.51	145.00	161.50	152.66	3.66	7.30	1.43
			10.28	12.33	4.72	10.49	19.95	54.12	2.02
4.	25	300	187.31	182.00	200.16	188.02	2.84	6.86	0.38
			11.54	13.34	5.01	12.00	15.56	56.62	3.98
5.	36	630	289.10	286.00	308.56	290.16	1.07	6.73	0.37
			14.98	16.16	6.01	15.37	7.83	59.92	2.58
6.	40	780	300.58	292.00	323.54	298.10	2.85	7.64	-0.83
			15.10	16.79	6.14	15.95	11.22	59.34	5.63
7.	45	990	359.76	354.00	385.05	358.82	1.60	7.03	-0.26
			17.02	18.22	5.70	17.45	7.08	66.53	2.56
8.	50	1225	408.12	402.00	436.80	406.57	1.50	7.03	-0.38
			17.37	19.18	5.66	18.51	10.43	67.45	6.53
9.	55	1485	418.43	409.00	449.98	413.43	2.25	7.54	-1.20
			18.09	20.07	5.47	19.47	10.94	69.78	7.63
10.	57	1596	444.37	435.00	478.27	440.00	2.11	7.63	-1.00
			18.50	20.52	5.61	19.91	10.94	69.97	7.63

TABLE 4.5: Results from NG 2 (5, 50)

Sl. No.	n	m	Simulation	Conventional	Sculli	PEA	Percent Deviations from Simulated Values: Conventional	Sculli	PEA
1.	5	10	97.82 8.31	97.00 8.89	98.69 7.59	97.65 8.35	0.84 6.99	0.89 8.67	-0.28 0.52
2.	15	105	356.26 13.74	342.00 15.65	368.72 8.60	352.79 13.58	4.00 13.92	3.50 37.43	-0.97 -1.17
3.	21	210	545.77 18.55	530.00 20.64	568.12 8.57	543.51 18.09	2.89 11.28	4.10 53.80	-0.41 -2.47
4.	25	300	710.59 21.55	702.00 22.67	731.19 11.51	712.30 20.60	1.21 5.22	2.90 46.56	0.24 -4.39
5.	36	630	1139.54 29.17	1134.00 30.40	1171.23 16.70	1141.87 29.16	0.49 4.19	2.78 42.76	0.21 -0.06
6.	40	780	1135.97 29.34	1124.00 30.79	1173.65 16.26	1133.64 29.42	1.05 4.92	3.32 44.60	-0.21 0.26
7.	45	990	1415.15 33.58	1405.00 33.99	1458.48 14.28	1414.30 32.66	0.72 1.22	3.06 57.47	-0.06 -2.74
8.	50	1225	1649.00 34.36	1643.00 36.11	1699.71 13.16	1650.07 35.04	0.36 5.09	3.08 61.70	0.06 1.99
9.	55	1485	1556.92 34.33	1543.00 36.61	1607.13 12.52	1553.77 34.96	0.89 6.63	3.23 63.53	-0.20 1.85
10.	57	1596	1736.80 35.22	1726.00 36.25	1791.70 14.73	1734.91 35.41	0.62 3.20	3.16 58.17	-0.11 0.53

TABLE 4.6: Results from NG 2 (1, 100)

Sl. No.	n		Simulation	Conventional	Sculli	PEA	Percent Deviations from Simulated Values: Conventional	Sculli	PEA
1.	5	10	176.47 10.86	175.00 11.58	177.36 9.92	175.77 11.03	0.83 6.57	0.50 8.56	-0.40 1.57
2.	15	105	655.34 16.69	645.00 18.81	669.06 11.07	656.69 15.91	1.58 12.73	2.09 33.65	0.21 -4.68
3.	21	210	1019.74 25.28	1010.00 28.05	1046.61 12.67	1024.97 24.43	0.96 10.97	2.63 49.89	0.51 -3.35
4.	25	300	1347.24 29.98	1343.00 31.35	1370.71 17.54	1349.77 29.70	0.31 4.59	1.74 41.50	0.19 -0.93
5.	36	630	2158.89 39.13	2150.00 41.10	2199.45 25.07	2159.88 39.62	0.41 5.02	1.88 35.95	0.05 1.25
6.	40	780	2150.56 39.62	2140.00 40.51	2192.76 [illegible]5.29	2151.55 39.34	0.49 2.25	1.96 36.17	0.05 -0.72
7.	45	990	2675.54 45.99	2669.00 46.20	2725.59 21.45	2676.19 44.77	0.24 0.46	1.87 53.36	0.02 -2.65
8.	50	1225	3147.51 46.94	3141.00 49.02	3206.82 19.37	3148.70 47.69	0.21 4.43	1.88 58.74	0.04 1.60
9.	55	1485	2915.81 45.92	2895.00 49.52	2974.71 18.03	2908.23 47.62	-0.71 7.84	2.[illegible]2 6[illegible].73	-0.26 3.70
10.	57	1596	3291.42 47.74	3285.00 49.19	3351.65 22.63	3292.78 48.17	0.20 3.05	1.83 52.59	0.04 0.91

TABLE 4.7: Results from NG 3 (5, 10)

Sl. No.	Network Dimensions W	L	n	m	Simulation	Conventional	Sculli	PEA	Percent Deviations from Simulated Values: Conventional	Sculli	PEA
1.	3	5	17	50	68.67 5.56	64.00 7.14	70.34 4.33	69.43 5.15	6.81 28.55	2.43 22.01	1.17 -7.26
2.	3	10	32	200	131.69 8.18	121.00 10.25	137.54 4.99	129.28 8.56	8.12 25.25	4.44 39.03	-1.83 4.66
3.	3	15	47	500	203.79 11.50	192.00 13.82	217.48 4.46	202.07 11.66	5.79 20.18	6.72 61.75	-0.85 1.43
4.	3	20	62	1000	334.44 14.55	320.00 17.35	357.87 5.56	329.41 15.81	4.32 19.26	7.01 61.78	-1.50 8.69
5.	4	10	42	550	205.39 11.05	193.00 13.38	218.47 11.73	202.52 11.73	6.03 21.10	6.37 57.66	-1.40 6.20
6.	7	5	37	150	82.84 5.50	75.00 7.81	85.36 3.51	82.36 5.61	9.46 41.88	3.05 36.22	-0.57 1.88
7.	8	10	82	900	237.77 11.14	220.00 14.42	249.37 4.40	232.32 11.74	7.47 29.47	4.88 60.53	-2.29 5.41
8.	8	12	98	1200	276.72 12.28	262.00 14.83	291.76 4.60	273.52 12.46	5.32 20.75	5.43 62.59	-1.16 5.43
9.	9	9	83	1150	265.62 12.36	252.00 14.93	278.48 5.20	263.22 13.03	4.75 20.83	4.84 57.89	-0.90 5.45
10.	10	5	52	400	146.10 9.12	134.00 11.75	153.35 4.96	143.66 9.68	8.28 28.86	4.97 45.56	-1.67 6.23

TABLE 4.8: Results from NG 3 (5, 50)

Sl. No.	Network Dimensions W	L	n	m	Simulation	Conventional	Sculli	PEA	Percent Deviations from Simulated Values: Conventional	Sculli	PEA
1.	3	5	17	50	273.42 12.63	271.00 14.04	275.74 11.25	273.56 12.76	0.88 11.15	0.85 10.90	0.05 1.08
2.	3	10	32	200	541.47 18.79	536.00 20.35	547.60 15.48	541.99 18.48	1.01 8.29	1.13 17.60	0.10 -1.64
3.	3	15	47	500	763.27 24.13	749.00 26.91	793.09 9.79	761.14 24.37	1.87 11.53	3.91 59.43	-0.28 1.00
4.	3	20	62	1000	1281.81 29.80	1272.00 32.39	1319.13 13.08	1281.22 30.64	0.77 8.67	2.91 56.11	-0.05 2.80
5.	4	10	42	550	762.74 21.80	751.00 23.47	784.25 13.28	764.82 20.88	1.54 7.70	2.82 39.06	0.27 -4.21
6.	7	5	37	150	334.61 10.89	332.00 12.81	338.62 8.36	334.80 10.87	0.78 17.54	1.20 23.31	0.06 -0.25
7.	8	10	82	900	860.38 22.06	842.00 27.40	880.62 9.84	855.56 24.51	2.11 24.25	2.35 55.38	-0.56 11.15
8.	8	12	98	1200	1047.52 23.16	1036.00 24.92	1070.50 11.09	1048.09 22.42	1.10 7.62	2.19 52.11	0.05 -3.20
9.	9	9	83	1150	1034.45 25.86	1027.00 27.89	1054.34 14.07	1036.92 25.55	0.72 7.85	1.92 45.60	0.24 -1.20
10.	10	5	52	400	567.08 19.93	561.00 22.29	571.98 16.15	570.16 19.79	1.07 11.87	0.86 18.95	0.54 -0.71

TABLE 4.9: Results from NG 3 (1, 100)

Sl. No.	Network Dimensions W	L	n	m	Simulation	Conven-tional	Sculli	PEA	Percent Deviations from Simulated Values: Conven-tional	Sculli	PEA
1.	3	5	17	50	523.71 17.65	522.00 19.03	526.50 16.21	523.67 18.16	0.33 7.77	0.53 8.17	-0.01 2.86
2.	3	10	32	200	1051.86 26.41	1047.00 27.95	1058.50 23.71	1052.43 26.24	0.46 5.80	0.63 10.22	0.06 -0.68
3.	3	15	47	500	1434.36 33.35	1417.00 35.54	1471.81 14.27	1434.57 32.00	1.21 6.56	2.61 57.21	0.01 -4.04
4.	3	20	62	1000	2427.91 41.58	2421.00 43.84	2470.10 20.10	2429.94 41.86	0.28 5.44	1.74 51.65	0.08 0.68
5.	4	10	42	550	1451.85 30.85	1445.00 33.05	1473.85 20.91	1455.95 30.46	0.47 7.12	1.52 32.23	0.28 -1.27
6.	7	5	37	150	645.51 15.78	645.00 16.76	649.94 12.79	645.91 15.74	0.08 6.25	0.69 18.93	0.06 -0.25
7.	8	10	82	900	1598.46 29.87	1572.00 35.10	1623.95 14.35	1592.55 30.77	1.66 17.50	1.59 51.94	-0.37 3.00
8.	8	12	98	1200	1989.19 31.03	1980.00 32.62	2017.53 16.39	1991.66 29.80	0.46 5.12	1.42 47.17	0.12 -3.96
9.	9	9	83	1150	1970.38 34.98	1962.00 37.51	1991.48 21.67	1971.59 34.74	0.43 7.24	1.07 38.05	0.06 -0.67
10.	10	5	52	400	1086.83 27.97	1082.00 30.69	1089.37 26.02	1089.44 27.95	0.44 9.74	0.23 6.96	0.24 -0.07

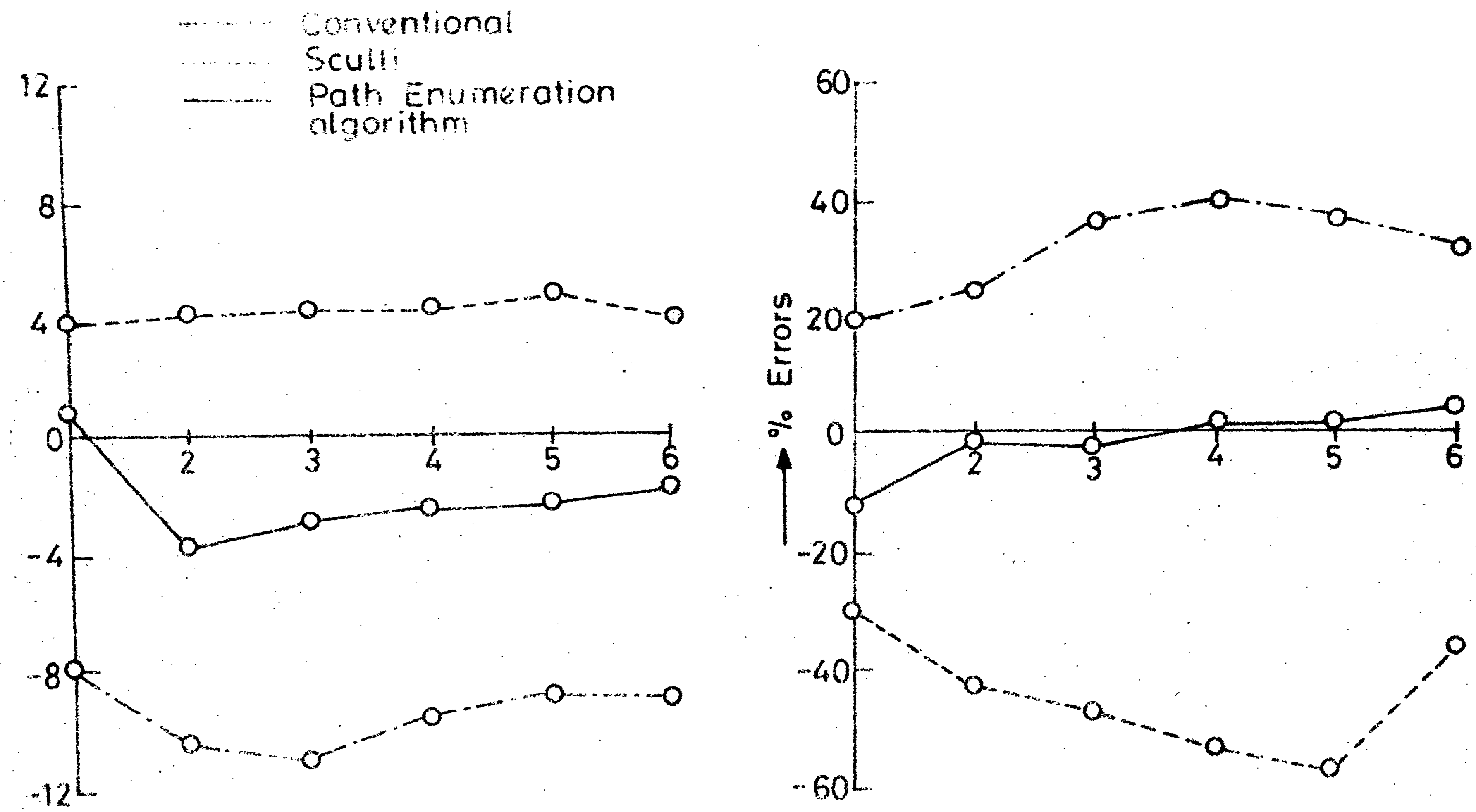

Fig. 4·5(A) Errors in means

Fig. 4·5(B) Errors in standard deviations.

Fig. 4·5 Percentage errors in various methods for the problems from NG 1

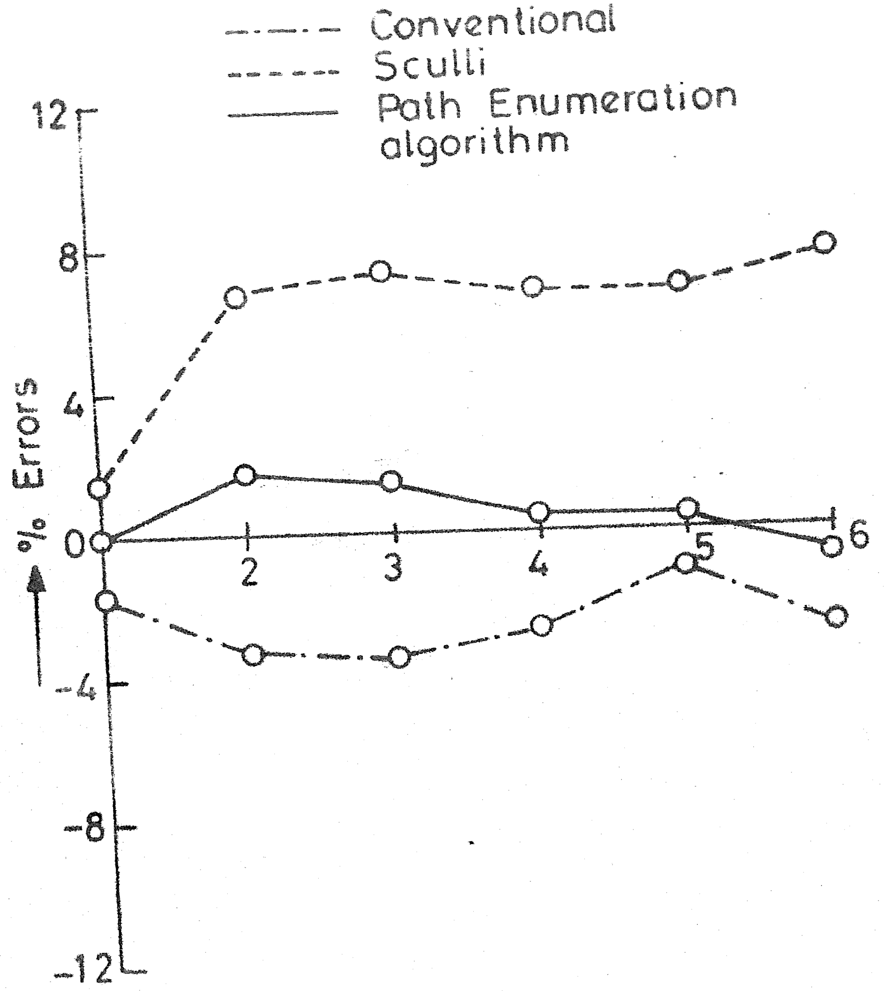

Fig. 4·6 (A) Errors in means

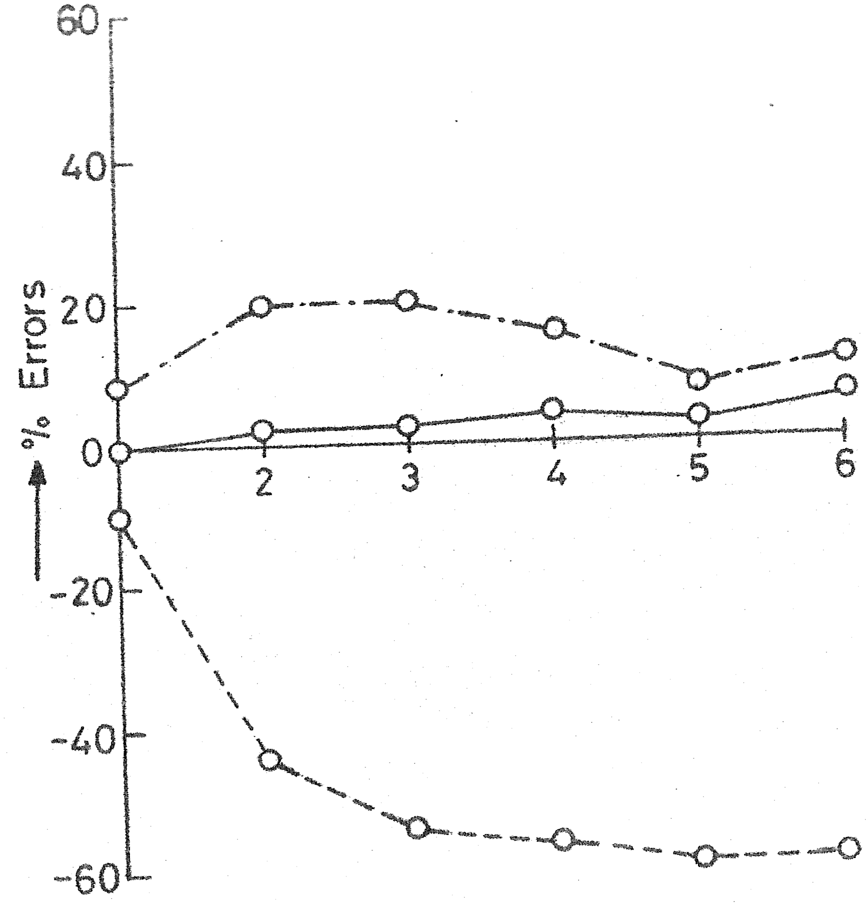

Fig. 4·6 (B) Errors in standard deviations

Fig. 4·6 Percentage errors in various methods for the problems from NG 2

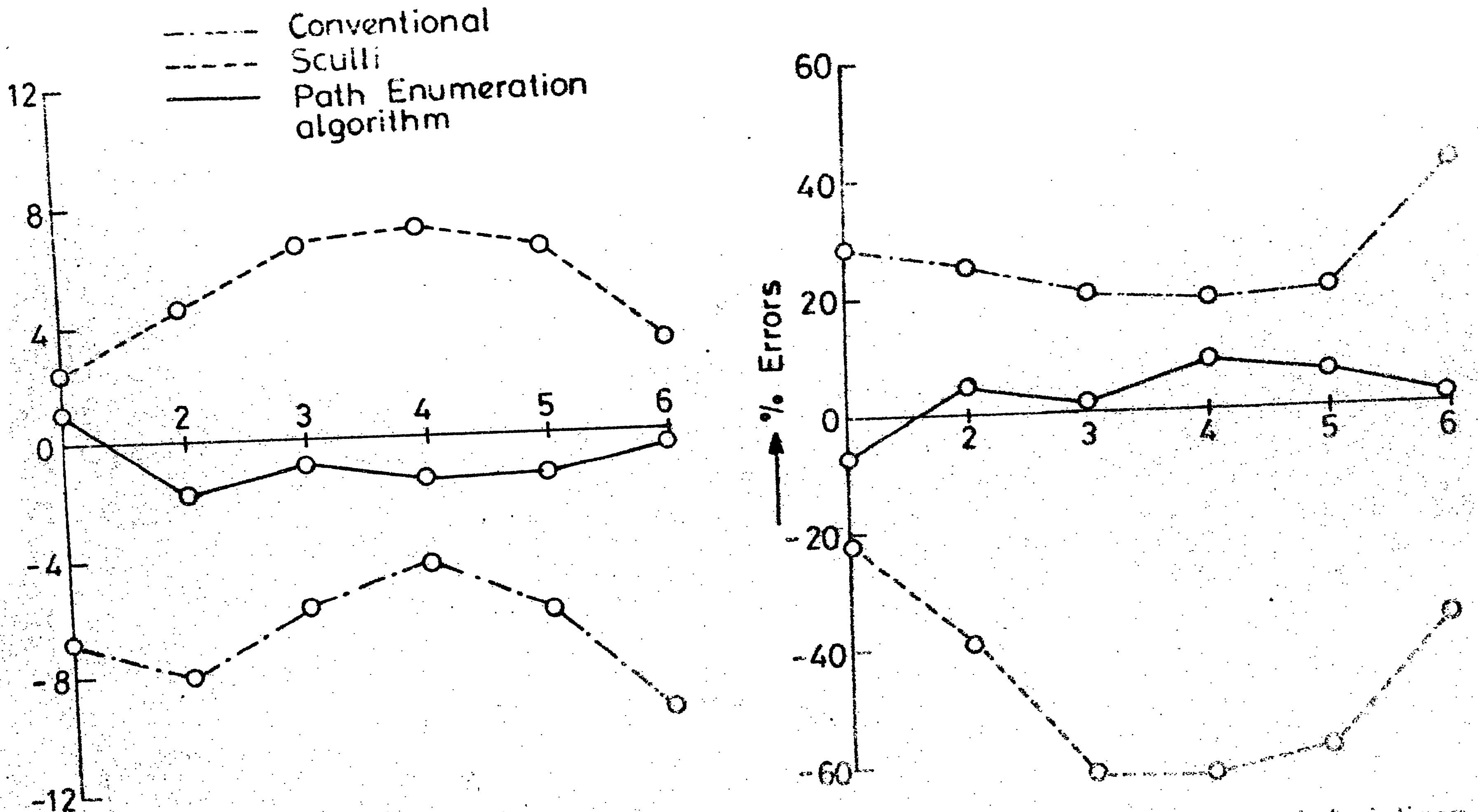

Fig.4·7(A) Errors in means

Fig.4·7(B) Errors in standard deviations

Fig. 4·7 Percentage errors in various methods for the problems from NG 3

TABLE 4.10: Execution times for PEA and Simulation (for NG 1).

S.No.	n	m	Average Execution Times				No. of Paths Enumerated								
			Simulation	PEA			5,10			5,50			1,100		
				5,10	5,50	1,100	Max.	Min.	Avg.	Max.	Min.	Avg.	Max.	Min.	Avg.
1.	42	111	11.95	0.26	0.18	0.12	32	26	30	24	10	17	21	11	14
2.	77	196	21.17	0.35	0.22	0.19	29	24	26	27	7	18	19	7	12
3.	102	261	28.15	0.57	0.25	0.22	40	32	38	28	7	14	32	7	13
4.	127	326	35.17	0.62	0.48	0.44	38	25	32	26	9	13	19	8	13
5.	162	441	47.48	4.37	3.59	3.18	36	26	31	18	13	16	26	9	14
6.	202	551	59.29	5.89	4.47	2.69	29	22	26	21	13	15	12	8	10
7.	242	661	71.23	6.81	5.26	3.57	27	20	23	23	9	13	11	7	9

TABLE 4.11: Execution times for PEA and Simulation (for NG 2).

S.No.	n	m	Average Execution Times				No. of Paths Enumerated								
			Simulation	PEA			5,10			5,50			1,100		
				5,10	5,50	1,100	Max.	Min.	Avg.	Max.	Min.	Avg.	Max.	Min.	Avg.
1.	25	300	31.65	0.30	0.32	0.21	28	21	24	27	19	24	13	9	10
2.	36	630	66.39	0.72	0.74	0.68	11	6	9	12	6	9	11	8	9
3.	40	780	82.12	4.24	4.42	3.07	25	5	19	26	17	21	13	7	11
4.	45	990	104.33	4.02	6.76	4.07	13	7	9	14	9	11	12	6	7
5.	50	1225	129.08	7.12	6.97	6.46	14	12	13	10	8	9	11	7	8
6.	55	1485	156.50	5.33	5.64	8.60	11	6	8	12	6	8	15	7	10
7.	57	1596	168.17	4.52	7.83	7.33	12	8	9	17	12	14	14	10	13

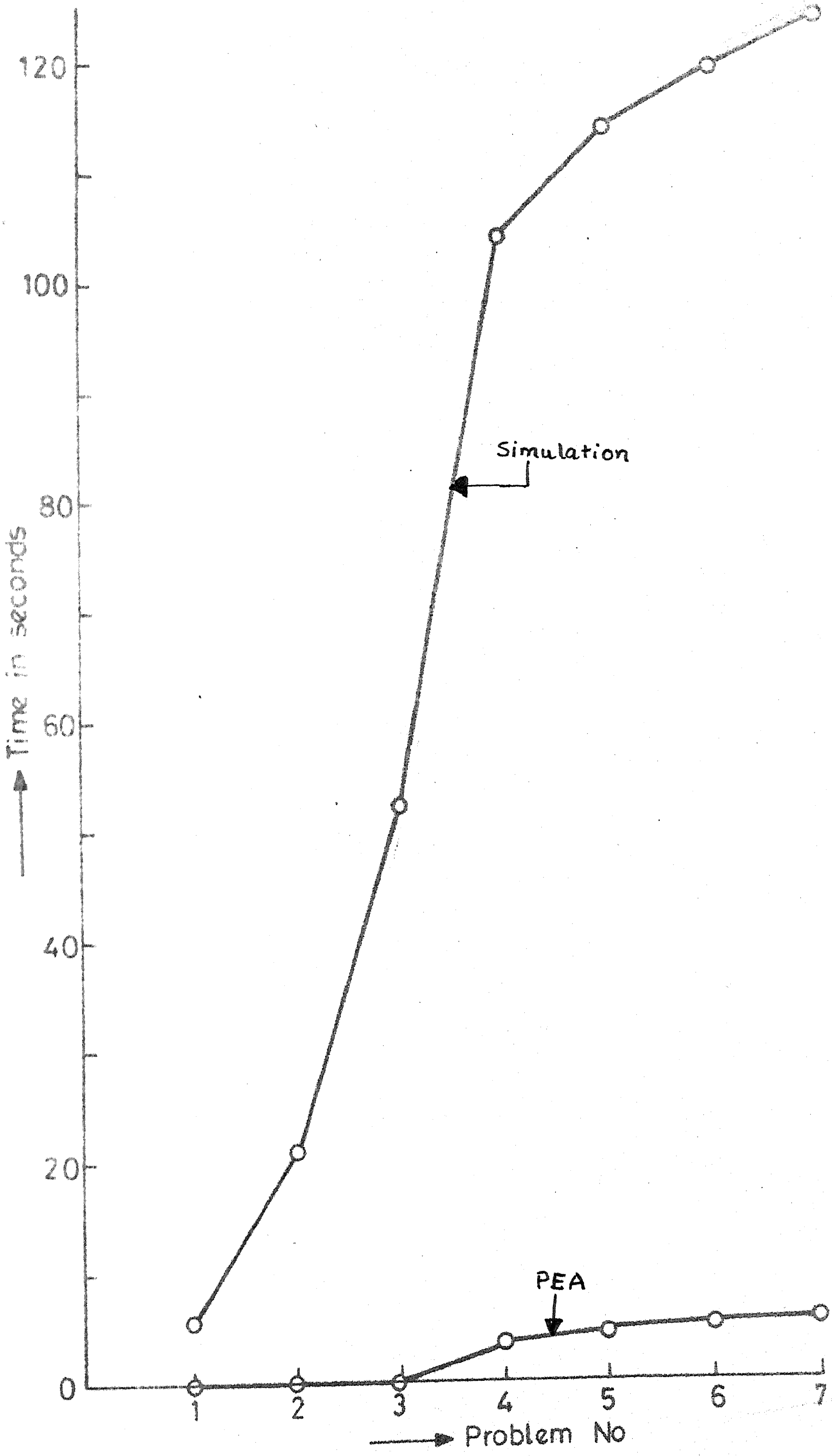

Fig. 4·8 Comparision of execution times

REFERENCES

1. Malcolm, D.G., Roseboom, J.H., Clark, C.E., and Fazar, W., Applications of a technique for R and D Program evaluation, (PERT) Operations Research, Vol. 7, No. 5, 1959, pp: 646-669.

2. Fulkerson, D.R., Expected critical path lengths in PERT networks, Operations Research, 10, 6, 1962, PP: 808-817.

3. Clingen, C.T., A Modification of Fulkerson s algorithm, Operations Research, 12, 4, 1964, PP: 629-632.

4. Elmaghraby, S.E., On the expected duration of PERT type networks, Management Science, 13, 1, 1967, PP: 299-306.

5. Lindsey, J.H., An estimate of expected critical path lengths in PERT networks, Operations Research, 20, 4, 1972, PP: 800-812.

6. Robillard, P. and Trahan, M., Expected completion time in PERT networks, Operations Research, 24, 1976, PP: 177-182.

7. Robillard, P. and Trahan, M., The Completion time of PERT networks, Operations Research, 25, 1977, PP: 15-29.

8. Sculli, D., The Completion time of PERT networks, Operations Research, 34, 2, 1983, PP: 155-158.

9. Greer, W.R., and Cacava, G.J., A Normal approximation for the timing of Stochastic events, Technical Report.

10. Fishman, G.S., Concepts and methods in discrete event digital simulation, John Wiley and Sons, 1973, pp: 211-213.

11. Naylor, T.H., J.L. Balintfy, Burdick, D.S., and Kongchu, Computer Simulation Techniques, John Wiley and Sons, New York, 1966, pp: 90-95.

12. Law, A.M., and Kelton, W.D., Simulation Modeling and Analysis, McGraw-Hill, 1982, pp: 258-259.

13. Clark, C.E., The greatest of a set of finite variables, Operations Research, 9, 1961, pp: 145-162.

14. Lawler, E.L., Combinatorial optimisation: Networks and Matroids, Holt, Rinehart and Winston, 1976, Pp: 98-100.

15. Rao, C.R., Linear Statistical Inference and its Applications, Wiley, New York, 1971, pp: 46.

16. Mac Crimmon, K.R., and Ryavec, C.A., An analytical study of the PERT assumptions, Operations Research, 12, 1964, pp: 16-37.

17. Martin, J.J., Distribution of time through a Directed Acyclic Network, Operations Research, 13, 1965, pp: 46-66.

18. Davies, R.B., Numerical inversion of a characteristic function, Biometrika, 60, 1973, pp: 415-417.

19. Pritsker, A.B., and Kiviat, P.J., Simulation with GASP II, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, pp: 198-201.

20. Kleindorfer, G.B., Bounding Distributions for a Stochastic Acyclic Network, Operations Research, 19, 1971, pp: 1586-1601.

```
      SUBROUTINE PEA
      REAL MEAN(1200),VAR(1200),ME,LABEL(100),MLAB(100),U(40,100),MEF,
     1 LARGE,V(40,100),M1DASH,M2DASH,M3DASH,JK,EMEAN(1200),EVAR(1200)
      INTEGER TAIL(1200),HEAD(1200),POINT(100),RPOINT(100),PARC(100),
     1 ARC(40,100),MU(40,1200),ARCS,THIRI,RLIST(1200),LABL(1200),
     2 COUNT
      COMMON NODES,ARCS,TAIL,HEAD,POINT,RPOINT,RLIST,MEAN,VAR,TSEED
      CALL RTIME(RTIME1)
      KOUNT=0
      ICOUNT=5
      KTH = 1
      CALL DIJ(KTH,LABEL,MLAB,ARC)
      EST1 = LABEL(NODES)
      VAR1 = MLAB(NODES)
      EPSLN=0.002*EST1
      COUNT=0
      N=NODES
 20   COUNT=COUNT+1
      IJ=ARC(KTH,N)
      PARC(COUNT)=IJ
      N=TAIL(IJ)
      LABL(IJ)=1
      IF(N.NE.1) GOTO 20
      DO 30 I=1,NODES
      U(1,I) = LABEL(I)
      V(1,I) = MLAB(I)
 30   CONTINUE
      DO 40 I=2,NODES
      J = ARC(KTH,I)
      MU(1,J) =1
 40   CONTINUE
      DO 50 I=1,ARCS
      EMEAN(I)=MEAN(I)
      EVAR(I)=VAR(I)
 50   CONTINUE
      KTH = 2
      LARGE = 10000.0
      VS=VAR1
 60   CALL REPEAT(KTH,U,V,ARC,MU,COUNT,PARC)
      TYPE *,KTH
      EST2 = U(KTH,NODES)
      VAR2 = V(KTH,NODES)
      IF(KTO.GE.ARCE) GOTO 136
```

```
      M3DASH=0.0
      V3DASH=0.0
      MK=0.0
      VK=0.0
      DO 70 I=COUNT,1,-1
      J=PARC(I)
      MK=MK+EMEAN(J)
      VK=VK+EVAR(J)
      IF(LABL(J).NE.1) GOTO 70
      M3DASH=M3DASH+EMEAN(J)
      V3DASH=V3DASH+EVAR(J)
70    CONTINUE
      V3DASH=V3DASH*VAR1/VS
      M1DASH=EST1-M3DASH
      M2DASH=MK-M3DASH
      IF(M2DASH.EQ.0.0) GOTO 120
      V1DASH=VAR1-V3DASH
      V2DASH=VK-V3DASH
      CALL MAX2(M1DASH,V1DASH,M2DASH,V2DASH,MEE,VEE)
      ME=MEE+M3DASH
      VE=VEE+V3DASH
C     IF(ABS(ME-EST1).GT.EPSLN) GOTO 80
C     KOUNT=KOUNT+1
      IF(KOUNT.GT.ICOUNT) GOTO 130
      GOTO 90
80    KOUNT=0
90    DO 110 I=COUNT,1,-1
      J=PARC(I)
      IF(LABL(J).EQ.1) GOTO 100
      VS=VS+EVAR(J)
      LABL(J)=1
100   EMEAN(J)=EMEAN(J)*EST1/ME
      EVAR(J)=EVAR(J)*VAR1/VE
110   CONTINUE
      EST1 = ME
      VAR1 = VE
120   KTH = KTH + 1
      IF(KTH.GT.40) GOTO 130
      GOTO 60
130   SD=SQRT(VE)
      CALL RTIME(NTIME2)
      THURI=NTIME2-NTIME1
      TYPE *,ME,SD
      WRITE(23,140) ME,SD,KTH
```

```
     1 'MEAN = ',F7.2,'   S.D. = ',F7.2,'   KTH = ',I3)
      WRITE(23,*),ME,SD
      WRITE(22,150),TMURI
  150 FORMAT(10X,'TIME FOR EXECUTION = ',I5,'  MILLISECONDS.')
      RETURN
      END
```

```
      SUBROUTINE REPEAT(KTH,U,V,ARC,MU,COUNT,PARC)
      INTEGER TAIL(1200),HEAD(1200),POINT(100),RPOINT(100),
     1 ARC(40,100),MU(40,1200),ARCS,RLIST(1200),COUNT,PARC(100),
     2 X,Y,STORE1,STORE2
      REAL VAR(1200),MEAN(1200),LABEL(100),MAP(100),U(40,100),MAX,
     1 SMALL,Z,LARGE,V(40,100)
      COMMON NODES,ARCS,TAIL,HEAD,POINT,RPOINT,RLIST,MEAN,VAR,ISEED
      SMALL = 0.0
      LARGE = 10000.0
      DO 10 I=1,NODES
      ARC(KTH,I)=ARC(KTH-1,I)
 10   CONTINUE
      DO 20 J=1,ARCS
      MU(KTH,J) = MU(KTH-1,J)
 20   CONTINUE
      U(KTH,1)=LARGE
      V(KTH,1)=SMALL
      DO 50 L=2,NODES
      IF(U(KTH-1,L).GE.LARGE) GOTO 30
      U(KTH,L)=LARGE
      GOTO 50
 30   U(KTH,L) = SMALL
      V(KTH,L) = SMALL
      IX = RPOINT(L)
      IY = RPOINT(L+1) -1
      IF(IX.GT.IY) GOTO 50
      MAX = 0.0
      DO 40 IZ = IX,IY
      X = RLIST(IZ)
      Y = TAIL(X)
      Z = U(MU(KTH-1,X)+1,Y) + MEAN(X)
      IF(Z.GE.LARGE) GOTO 40
      IF(Z.LT.MAX) GOTO 40
      MAX = Z
      STORE1 = X
      STORE2 = Y
 40   CONTINUE
      IF(MAX.EQ.SMALL) MAX = LARGE
      U(KTH,L) = MAX
      IF(MAX.EQ.LARGE) GOTO 50
      V(KTH,L) = V(MU(KTH-1,STORE1)+1,STORE2) + VAR(STORE1)
      MU(KTH,STORE1) = MU(KTH,STORE1) + 1.
      IPC(KTH,L) = STORE1
```

```
50    CONTINUE
      I=NODES
      COUNT=0
      PATH=KTH
60    COUNT=COUNT+1
      IJ=ARC(PATH,I)
      PARC(COUNT)=IJ
      I=TAIL(IJ)
      PATH=MU(PATH,IJ)
      IF(I.NE.1) GOTO 60
      RETURN
      END
```

```
C       DIJKSTRA'S ALGORITHM.

        SUBROUTINE DIJ(KTH,LABEL,MLAB,ARC)
        INTEGER TAIL(1200),HEAD(1200),POINT(100),RPOINT(100),
        1 ARC(40,100),NU(40,1200),ARCS,RLIST(1200),COUNT,PARC(100)
        REAL VAR(1200),MEAN(1200),LABEL(100),MLAB(100),U(40,100),MAX,
        1 NLABEL,V(40,100)
        COMMON NODES,ARCS,TAIL,HEAD,POINT,RPOINT,RLIST,MEAN,VAR,ISEED
        ARC(KTH,1) = 0
        DO 10 I=1,NODES
        LABEL(I) = 0.0
        MLAB(I) = 0.0
   10   CONTINUE
        DO 30 I=1,NODES-1
        IX = POINT(I)
        IY = POINT(I+1) - 1
        DO 20 IL=IX,IY
        LHEAD = HEAD(IL)
        LTAIL = TAIL(IL)
        NLABEL = LABEL(LTAIL) + MEAN(IL)
        IF(NLABEL.LE.LABEL(LHEAD)) GOTO 20
        LABEL(LHEAD) = NLABEL
        MLAB(LHEAD) = MLAB(LTAIL) + VAR(IL)
        ARC(KTH,LHEAD) = IL
   20   CONTINUE
   30   CONTINUE
        RETURN
        END
```

```
C      VARIABLES USING CLARK'S  FORMULAE.

       SUBROUTINE MAX2(EST1,VAR1,EST2,VAR2,MUE,VEE)
       REAL MUE,MOMENT
       SD1=SQRT(VAR1)
       SD2=SQRT(VAR2)
       A=SQRT(SD1**2+SD2**2)
       ALPHA = (EST1-EST2)/A
       CALL PHISI(ALPHA,PHI1,SI)
       PHI2 = 1.0-PHI1
       MUE = EST1*PHI1+EST2*PHI2+A*SI
       MOMENT=(EST1**2+SD1**2)*PHI1+(EST2**2+SD2**2)*PHI2+
      1 (EST1+EST2)*A*SI
       VEE = MOMENT-MUE**2
       RETURN
       END

C      THIS SUBROUTINE GIVES PHI AND SI VALUES FOR A GIVEN X.

       SUBROUTINE PHISI(X,C,D)
       X1=ABS(X)
       T=1.0/(1.0+0.2316419*X1)
       D=0.3989423*EXP(-X*X/2.0)
       C=(1.330274*T-1.821256)*T+1.781478
       C=1.0-D*T*((C*T-0.3565638)*T+0.3193815)
       IF(X.GT.0.0) RETURN
       C=1.0-C
       RETURN
       END
```

```
C          AND SIMULATION APPROACHES.

      SUBROUTINE SIMU
      INTEGER ARCS,TAIL(1200),HEAD(1200),POINT(100),RUNS,RPOINT(100),
     1 RLIST(1200),TCONV,TSIMU
      REAL MEAN(1200),VAR(1200),TIME1(1200),TIME2(1200)
      COMMON NODES,ARCS,TAIL,HEAD,POINT,RPOINT,RLIST,MEAN,VAR,ISEED
      CALL RTIME(NTIME1)
      CALL PROJ02(AVRAGE,VARI)
      SDD = SQRT(VARI)
      CALL RTIME(NTIME2)
      TCONV=NTIME2-NTIME1
      TYPE *,AVRAGE,SDD
      WRITE(22,10),AVRAGE,SDD
10    FORMAT(10X,'RESULTS OF CONVENTIONAL METHOD :',//,10X,'MEAN =
     1 F7.2,'  S.D.= ',F7.2)
      WRITE(23,*),AVRAGE,SDD
      WRITE(22,20),TCONV
20    FORMAT(10X,'TIME FOR EXECUTION =',I4,'  MILLISECONDS.')
      CALL RTIME(NTIME1)
      TYPE 30
30    FORMAT(10X,'ENTER THE NO. OF RUNS TO BE MADE : ',$)
      ACCEPT *,RUNS
      CALL SETRAN(ISEED)
      SUM = 0.0
      SUMSQ = 0.0
      DO 50 I=1,RUNS,2
      DO 40 J=1,ARCS
      A = MEAN(J)
      B = SQRT(VAR(J))
      CALL RAN(A,B,X1,X2)
      TIME1(J) = X1
      TIME2(J) = X2
40    CONTINUE
      CALL PROJ01(TIME1,TERM1)
      CALL PROJ01(TIME2,TERM2)
      SUM = SUM+TERM1+TERM2
      SUMSQ = SUMSQ+TERM1**2+TERM2**2
50    CONTINUE
      PMEAN = SUM/RUNS
      PSD   = SQRT(SUMSQ/RUNS-PMEAN**2)
      CALL RTIME(NTIME2)
      TSIMU=NTIME2-NTIME1
```

```
      WRITE(22,60),PMEAN,PSD
60    FORMAT(//,10X,'RESULTS OF SIMULATION APPROACH :',//,10X,
     1 'MEAN = ',F7.2,'  S.D.= ',F7.2)
      WRITE(23,*),PMEAN,PSD
      WRITE(22,70),TSIM
70    FORMAT(10X,'TIME FOR EXECUTION = ',I7,'  MILLISECONDS.')
      RETURN
      END

C     SUBROUTINE USED IN SIMULATION APPROACH.

      SUBROUTINE PROJO1(TIME,AVRAGE)
      INTEGER ARCS,X,Y,Z,POINT(100),TAIL(1200),HEAD(1200),RPOINT(100),
     1 RLIST(1200)
      REAL LABEL(100),MEAN(1200),VAR(1200),TIME(1200)
      COMMON NODES,ARCS,TAIL,HEAD,POINT,RPOINT,RLIST,MEAN,VAR,ISEED
      DO 10 I=1,NODES
      LABEL(I) = 0.0
10    CONTINUE
      DO 30 J=1,NODES
      X = POINT(J)
      Y = POINT(J+1) - 1
      IF(Y.LT.X) GOTO 40
      DO 20 K = X,Y
      Z = HEAD(K)
      LABEL(Z) = AMAX1(LABEL(Z),LABEL(J)+TIME(K))
20    CONTINUE
30    CONTINUE
40    AVRAGE = LABEL(NODES)
      RETURN
      END

C     SUBROUTINE USED IN CONVENTIONAL METHOD.

      SUBROUTINE PROJO2(AVRAGE,VARI)
      INTEGER ARCS,X,Y,Z,POINT(100),HEAD(1200),TAIL(1200),RPOINT(100),
     1 RLIST(1200)
      REAL LABEL(100),MEAN(1200),VAR(1200),MLAB(100),MLABEL
      COMMON NODES,ARCS,TAIL,HEAD,POINT,RPOINT,RLIST,MEAN,VAR,ISEED
      DO 10 I=1,NODES
      LABEL(I) = 0.0
      MLAB(I)  = 0.0
10    CONTINUE
      DO 30 J=1,NODES
```

```
      Y = POINT(J+1) - 1
      IF(Y.LT.X) GOTO 40
      DO 20 K = X,Y
      Z = HEAD(K)
      NLABEL = LABEL(J) + MEAN(K)
      IF(NLABEL.LE.LABEL(Z)) GOTO 20
      LABEL(Z) = NLABEL
      NLAB (Z) = NLAB(J) + VAR(K)
 20   CONTINUE
 30   CONTINUE
 40   AVRAGE = LABEL(NODES)
      VARI   = NLAB (NODES)
      RETURN
      END

C     SUBROUTINE FOR GENERATING NORMAL VARIATES.

      SUBROUTINE RANN(AVRAGE,SD,X1,X2)
 10   U1 = RAN(X)
      U2 = RAN(X)
      V1 = 2.0*U1-1.0
      V2 = 2.0*U2-1.0
      W  = V1*V1+V2*V2
      IF(W.GT.1.0) GOTO 10
      Y = SQRT(-2.0*ALOG(W)/W)
      XI = V1*Y
      XII = V2*Y
      X1 = XI*SD+AVRAGE
      X2 = XII*SD+AVRAGE
      IF(X1.LT.0.0) X1=0.0
      IF(X2.LT.0.0) X2=0.0
      RETURN
      END
```

```
      SUBROUTINE SCULLI
      INTEGER ARCS,TAIL(1200),HEAD(1200),POINT(100),RPOINT(100),
     1 RLIST(1200),TSCUL
      REAL MEAN(1200),VAR(1200),LABEL(100),MLAB(100),MU1,MU2,MUE
      COMMON NODES,ARCS,TAIL,HEAD,POINT,RPOINT,RLIST,MEAN,VAR,ISEED
      CALL RTIME(NTIME1)
      LABEL(1) = 0.0
      MLAB(1) = 0.0
      DO 30 I=2,NODES
      LABEL(I)=0.0
      MLAB(I)=0.0
      IX=RPOINT(I)
      IY=RPOINT(I+1)-1
      IF(IX.GT.IY) GOTO 30
      J=RLIST(IX)
      MU1=LABEL(TAIL(J))+MEAN(J)
      VAR1=MLAB(TAIL(J))+VAR(J)
      IF(IY.GT.IX) GOTO 10
      MUE=MU1
      VARE=VAR1
      GOTO 20
10    IX=IX+1
      J=RLIST(IX)
      MU2=LABEL(TAIL(J))+MEAN(J)
      VAR2=MLAB(TAIL(J))+VAR(J)
      CALL MAX2(MU1,VAR1,MU2,VAR2,MUE,VARE)
      IF(IX.EQ.IY) GOTO 20
      MU1=MUE
      VAR1=VARE
      GOTO 10
20    LABEL(I)=MUE
      MLAB(I) =VARE
30    CONTINUE
      ESTMTE = LABEL(NODES)
      SD = SQRT(MLAB(NODES))
      CALL RTIME(NTIME2)
      TSCUL=NTIME2-NTIME1
      TYPE *,ESTMTE,SD
      WRITE(22,40),ESTMTE,SD
40    FORMAT(//,10X,'RESULTS OF SCULLI''S METHOD :',//,10X,'MEAN =
     1 F7.2,'  S.D. = ',F7.2)
      WRITE(23,*),ESTMTE,SD
```

```
50    FORMAT(10X,'TIME FOR EXECUTION = ',I4,' MILLISECONDS.')
      RETURN
      END
```

```
C       SIMULATED VALUES.

        SUBROUTINE RATIO
        REAL MECONV,MESIMU,MEHURI,MESCUL
        READ(23,*),MECONV,SDCONV,MESIMU,SDSIMU,MEHURI,SDHURI,
        1 MESCUL,SDSCUL
        R1CONV=MECONV/MESIMU
        R2CONV=SDCONV/SDSIMU
        R1HURI=MEHURI/MESIMU
        R2HURI=SDHURI/SDSIMU
        R1SCUL=MESCUL/MESIMU
        R2SCUL=SDSCUL/SDSIMU
        WRITE(22,10),R1CONV,R2CONV,R1HURI,R2HURI,R1SCUL,R2SCUL
10      FORMAT(//,15X,6F8.4)
        R1PC=ABS(1.0-R1CONV)*100.0
        R2PC=ABS(1.0-R2CONV)*100.0
        R1PH=ABS(1.0-R1HURI)*100.0
        R2PH=ABS(1.0-R2HURI)*100.0
        R1PS=ABS(1.0-R1SCUL)*100.0
        R2PS=ABS(1.0-R2SCUL)*100.0
        WRITE(22,15),R1PC,R2PC,R1PH,R2PH,R1PS,R2PS
15      FORMAT(15X,6F8.2)
        RETURN
        END
```

```
      INTEGER ARCS,TAIL(900),HEAD(900),POINT(300),RPOINT(300),
     1 RLIST(900)
      REAL MEAN(900),VAR(900)
      COMMON NODES,ARCS,TAIL,HEAD,POINT,RPOINT,RLIST,MEAN,VAR,ISEED
      CALL NG1
      CALL SIMU
      CALL PEA
      CALL SCULLI
      OPEN(UNIT=23,MODE='ASCII')
      CALL RATIO
      CLOSE(UNIT=23,MODE='ASCII')
      STOP
      END

C     THIS SUBROUTINE GENERATES NETWORKS OF TYPE-1.

      SUBROUTINE NG1
      INTEGER WIDTH,ARCS,COUNT,TAIL(900),HEAD(900),POINT(300),
     1 ALIST(300),BLIST(300),RPOINT(300),RLIST(900)
      REAL MEAN(900),VAR(900)
      COMMON NODES,ARCS,TAIL,HEAD,POINT,RPOINT,RLIST,MEAN,VAR,ISEED
      TYPE 10
   10 FORMAT(10X,'ENTER THE WIDTH REQUIRED : ',$)
      ACCEPT *,WIDTH
      TYPE 20
   20 FORMAT(10X,'ENTER THE LENGTH REQUIRED : ',$)
      ACCEPT *,LENGTH
      CALL SETRAN(ISEED)
      NODES=LENGTH*WIDTH+2
      COUNT=0
      DO 30 I=2,WIDTH+1
      COUNT=COUNT+1
      TAIL(COUNT)=1
      HEAD(COUNT)=I
   30 CONTINUE
      DO 50 K=1,LENGTH-1
      IS=(K-1)*WIDTH+2
      IT=IS+WIDTH-1
      DO 50 I=IS,IT
      IF(I.EQ.IT)GO TO 40
      COUNT=COUNT+1
```

```
      HEAD(COUNT)=I+1
   40 COUNT=COUNT+1
      TAIL(COUNT)=I
      HEAD(COUNT)=I+WIDTH
      IF(I.EQ.IT)GO TO 50
      COUNT=COUNT+1
      TAIL(COUNT)=I
      HEAD(COUNT)=I+WIDTH+1
   50 CONTINUE
      IS=IS+WIDTH
      IT=IT+WIDTH
      DO 70 I=IS,IT
      IF(I.EQ.IT)GO TO 60
      COUNT=COUNT+1
      TAIL(COUNT)=I
      HEAD(COUNT)=I+1
   60 COUNT=COUNT+1
      TAIL(COUNT)=I
      HEAD(COUNT)=NODES
   70 CONTINUE
      ARCS=COUNT
      DO 80 I=1,NODES
      ALIST(I)=0
   80 CONTINUE
      POINT(1)=1
      K=1
      DO 90 J=1,ARCS
      ALIST(HEAD(J)) = ALIST(HEAD(J))+1
      IF(TAIL(J).EQ.K)GO TO 90
      K=K+1
      POINT(K)=J
   90 CONTINUE
      POINT(NODES)=ARCS+1
      POINT(NODES+1)=ARCS+1
      BLIST(1) = 1
      RPOINT(1) = 1
      DO 100 I=2,NODES
      BLIST(I)=BLIST(I-1)+ALIST(I-1)
      RPOINT(I)=BLIST(I)
  100 CONTINUE
      RPOINT(NODES+1)=ARCS+1
      DO 110 J=1,ARCS
      MEAN(J)=RAND(5,10)
         R(J)=RAND(5,10)
```

```
      BLIST(HEAD(J))=BLIST(HEAD(J))+1
  110 CONTINUE
      WRITE(22,120),WIDTH,LENGTH,NODES,ARCS
  120 FORMAT(10X,'WIDTH = ',I3,' LENGTH = ',I3,'  NODES = ',I3,
     1 '  ARCS = ',I5,//)
      RETURN
      END


C     FUNCTION FOR GENERATING A UNIFORMLY DISTRIBUTED RANDOM NUMBER
C     BETWEEN TWO SPECIFIED INTEGERS.

      FUNCTION RAND(IC,ID)
      IRAN=IC+(ID-IC+1)*RAN(X)
      IF(IRAN.GT.ID)IRAN=ID
      RAND=IRAN
      RETURN
      END
```

```
      INTEGER ARCS,TAIL(1600),HEAD(1600),POINT(60),RPOINT(60),
     1 RLIST(1600)
      REAL MEAN(1600),VAR(1600)
      COMMON NODES,ARCS,TAIL,HEAD,POINT,RPOINT,RLIST,MEAN,VAR,ISEED
      CALL NG2
      CALL SIMU
      CALL PEA
      CALL SCULLI
      OPEN(UNIT=23,MODE='ASCII')
      CALL RATIO
      CLOSE(UNIT=23,MODE='ASCII')
      STOP
      END


C     THIS SUBROUTINE GENERATES NETWORKS OF TYPE-2.

      SUBROUTINE NG2
      INTEGER ARCS,TAIL(1600),HEAD(1600),COUNT,ALIST(60),BLIST(60),
     1 POINT(60),RPOINT(60),RLIST(1600)
      REAL MEAN(1600),VAR(1600)
      COMMON NODES,ARCS,TAIL,HEAD,POINT,RPOINT,RLIST,MEAN,VAR,ISEED
      TYPE 10
 10   FORMAT(10X,'ENTER THE NO. OF NODES : ',S)
      ACCEPT *,NODES
      COUNT=0
      N=NODES
      DO 20 I=1,N-1
      DO 20 K=I+1,N
      COUNT=COUNT+1
      TAIL(COUNT)=I
      HEAD(COUNT)=K
 20   CONTINUE
      ARCS=COUNT
      DO 30 I=1,NODES
      ALIST(I)=0
 30   CONTINUE
      K=1
      POINT(1)=1
      DO 40 J=1,ARCS
      ALIST(HEAD(J))=ALIST(HEAD(J))+1
      IF(TAIL(J).EQ.K)GO TO 40
```

```
      POINT(K)=J
   40 CONTINUE
      POINT(NODES)=ARCS+1
      POINT(NODES+1)=ARCS+1
      BLIST(1)=1
      RPOINT(1)=1
      DO 50 I=2,NODES
      BLIST(I)=BLIST(I-1)+ALIST(I-1)
      RPOINT(I)=BLIST(I)
   50 CONTINUE
      RPOINT(NODES+1)=ARCS+1
      DO 60 J=1,ARCS
      MEAN(J) = RAND(5,10)
      VAR(J)=RAND(5,10)
      RLIST(BLIST(HEAD(J)))=J
      BLIST(HEAD(J))=BLIST(HEAD(J))+1
   60 CONTINUE
      WRITE(22,70),NODES,ARCS
   70 FORMAT(10X,'NODES = ',I3,'  ARCS = ',I5,//)
      RETURN
      END
```

```
      INTEGER ARCS,TAIL(1200),HEAD(1200),POINT(100),RPOINT(100),
     1 RLIST(1200)
      REAL MEAN(1200),VAR(1200)
      COMMON NODES,ARCS,TAIL,HEAD,POINT,RPOINT,RLIST,MEAN,VAR,ISEED
      CALL NG3
      CALL SIMU
      CALL PCA
      CALL SCULLI
      OPEN(UNIT=23,MODE='ASCII')
      CALL RATIO
      CLOSE(UNIT=23,MODE='ASCII')
      STOP
      END

C     THIS SUBROUTINE GENERATES NETWORKS OF TYPE-3.

      SUBROUTINE NG3
      INTEGER ARCS,WIDTH,TAIL(1200),HEAD(1200),POINT(100),RPOINT(100),
     1 RLIST(1200),COUNT,A(100,100),ALIST(100),BLIST(100)
      REAL MEAN(1200),VAR(1200)
      COMMON NODES,ARCS,TAIL,HEAD,POINT,RPOINT,RLIST,MEAN,VAR
      TYPE 10
10    FORMAT(10X,'ENTER THE WIDTH : ',$)
      ACCEPT *,WIDTH
      TYPE 20
20    FORMAT(10X,'ENTER THE LENGTH : ',$)
      ACCEPT *,LENGTH
      MIN=(LENGTH+1)*WIDTH
      NODES=WIDTH*LENGTH+2
      MAX=(NODES*(NODES-1))/2
      TYPE 30,MIN,MAX
30    FORMAT(10X,'ENTER THE NO.OF ARCS IN THE RANGE',I3,' - ',I4)
      TYPE 40
40    FORMAT(/,10X,'ENTER THE NO. OF ARCS : ',$)
      ACCEPT *,ARCS
      COUNT=0
      DO 50 J=2,WIDTH+1
      COUNT=COUNT+1
      A(1,J)=1
50    CONTINUE
      K=2
```

```
      DO 60 J=K,K+WIDTH-1
      COUNT=COUNT+1
      A(J,J+WIDTH)=1
 60   CONTINUE
      K=K+WIDTH
 70   CONTINUE
      DO 80 I=K,K+WIDTH-1
      COUNT=COUNT+1
      A(I,NODES)=1
 80   CONTINUE
 90   IF(COUNT.GE.ARCS) GOTO 140
100   NODE1=RAND(1,NODES)
110   NODE2=RAND(1,NODES)
      IF(NODE1-NODE2) 120,110,130
120   IF(A(NODE1,NODE2).EQ.1) GOTO 100
      COUNT=COUNT+1
      A(NODE1,NODE2)=1
      GOTO 90
130   IF(A(NODE2,NODE1).EQ.1) GOTO 100
      COUNT=COUNT+1
      A(NODE2,NODE1)=1
      GOTO 90
140   KOUNT=0
      DO 150 I=1,NODES
      DO 150 J=1,NODES
      IF(I.GE.J) GOTO 150
      IF(A(I,J).NE.1) GOTO 150
      KOUNT=KOUNT+1
      TAIL(KOUNT)=I
      HEAD(KOUNT)=J
150   CONTINUE
      IF(KOUNT.EQ.ARCS) GOTO 170
      TYPE 160
160   FORMAT(10X,'SOME WRONG IN THE PROGRAM.')
      STOP
170   DO 180 I=1,NODES
      ALIST(I)=0
180   CONTINUE
      K=1
      POINT(1)=1
      DO 190 J=1,ARCS
      ALIST(HEAD(J))=ALIST(HEAD(J))+1
      IF(TAIL(J).EQ.K)GO TO 190
      K=K+1
```

```
190   CONTINUE
      POINT(NODES)=ARCS+1
      POINT(NODES+1)=ARCS+1
      BLIST(1)=1
      RPOINT(1)=1
      DO 200 I=2,NODES
      BLIST(I)=BLIST(I-1)+ALIST(I-1)
      RPOINT(I)=BLIST(I)
200   CONTINUE
      RPOINT(NODES+1)=ARCS+1
      DO 210 J=1,ARCS
      MEAN(J) = RAND(5,10)
      VAR(J)=RAND(5,10)
      RLIST(BLIST(HEAD(J)))=J
      BLIST(HEAD(J))=BLIST(HEAD(J))+1
210   CONTINUE
      WRITE(22,220),WIDTH,LENGTH,NODES,ARCS
220   FORMAT(10X,'WIDTH = ',I3,'  LENGTH = ',I3,'  NODES = ',
     1 I4,'  ARCS = ',I5,//)
      RETURN
      END
```